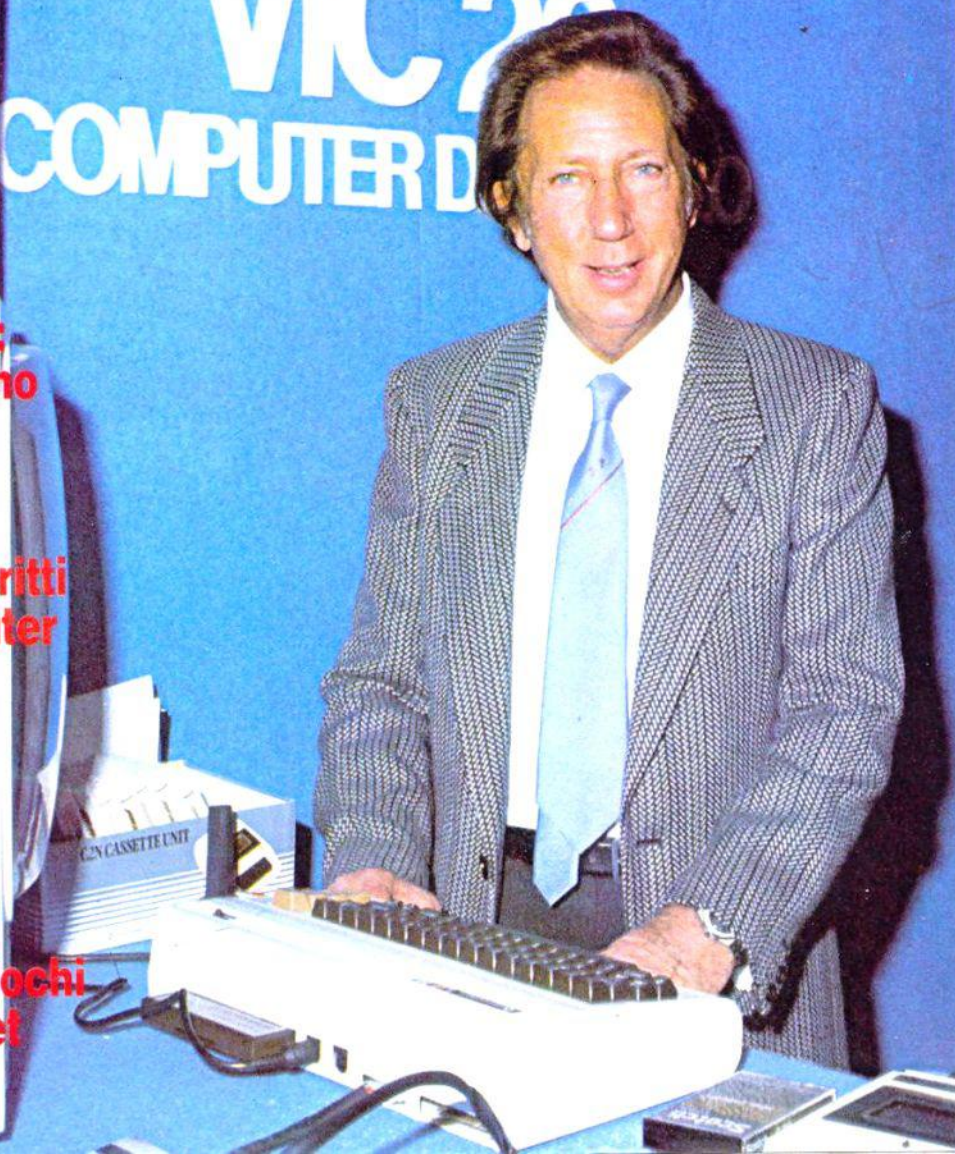
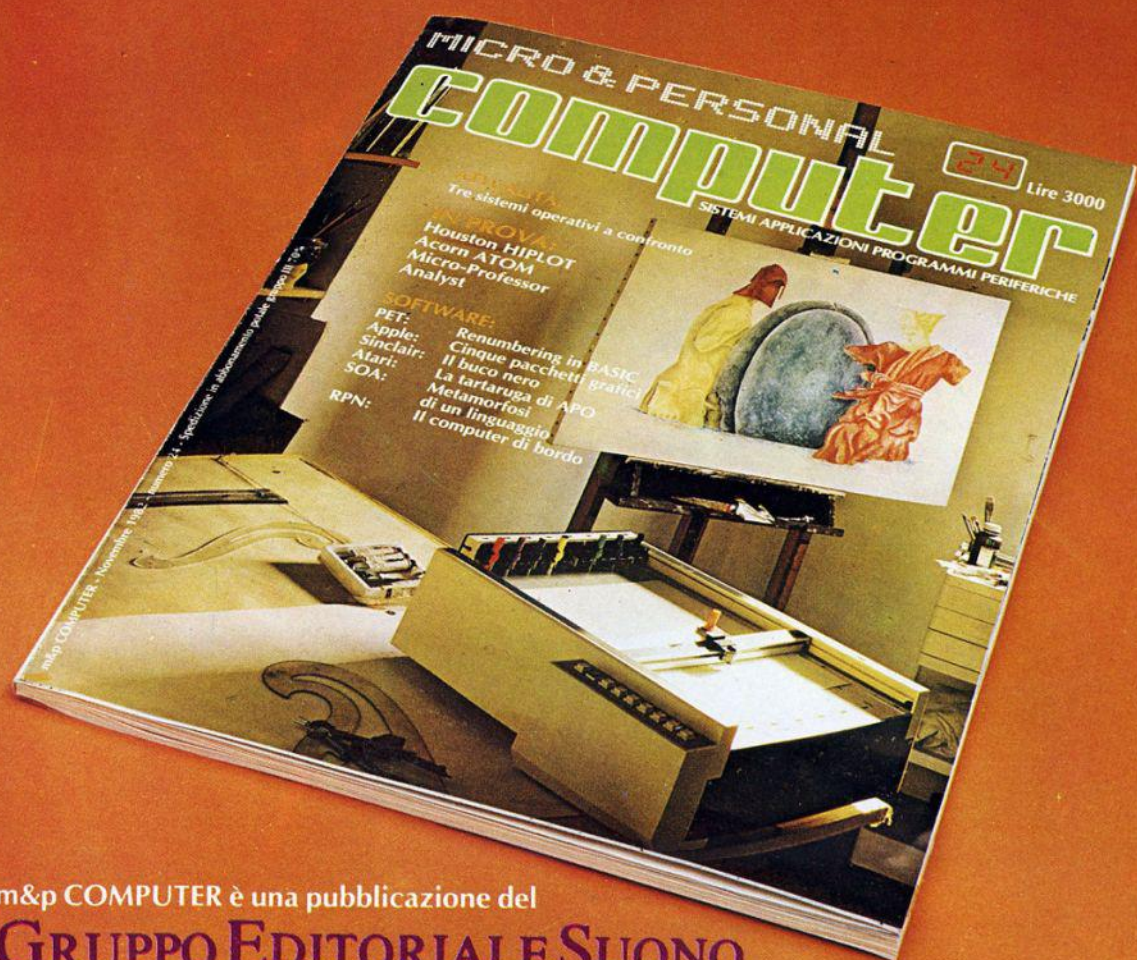
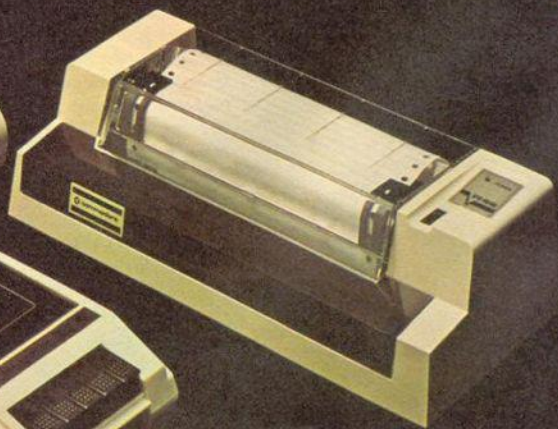
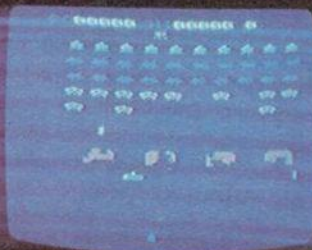
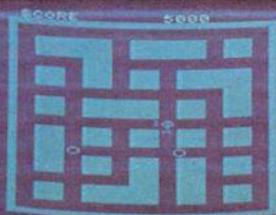
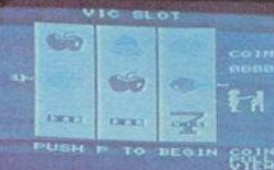
commodore
COMPUTER
CLUB
N. 3
Lire 2000
La rivista degli utenti di sistemi Commodore
Suppl. a "Computer" 56 - 28 febbr. 30 marzo 1983 Sped. Abb. post. gr. III/70
VIC 20
COMPUTER D
Commodoriani:
Mike Buongiorno
Come tradurre
i programmi scritti
per altri computer
Dichiarazione
dei redditi
Programmi e giochi
per il Vic e il Pet
C2N CASSETTE UNIT

Il "mago dei telegiochi", Mike Buongiorno, al suo primo incontro con il VIC. Il noto presentatore recentemente ha acquistato questo computer dalla Editrice Giochi, che è il distributore nazionale del VIC per il canale giocattoli.

SOMMARIO

Commodore Computer Club - Rivista indipendente per gli utenti di sistemi Commodore.
Direttore responsabile: Michele di Pisa
Redazione: Alessandro De Simone, Franco Rao
Direzione, redazione: Piazza Arduino, 3 - 20149 Milano - Tel. (02) 434354-435717
Pubblicità: Milano: Paola Bevilacqua, Gianluigi Centurelli, Mario Gandolfo, Tina Ronchetti, Villa Claudio. - Piazza Arduino, 3 - 20149 Milano - Tel. (02) 434354
Prezzi e abbonamenti: la rivista esce bimestralmente. Prezzo per una copia Lire 2.000. Arretrati il doppio. Abbonamento per dieci fascicoli lire 18.000. Abbonamento annuo cumulativo alle riviste Computer e Commodore Computer Club (tariffa riservata agli studenti): L. 24.000. I versamenti vanno indirizzati a: Minisystems Italia s.r.l., mediante assegno bancario; vaglia o utilizzando il c/c postale n. 11909207.
Composizioni: Minisystems Italia
Selezioni: Org. Aldo Ghiacci
Stampa:
Registrazione: Tribunale di Milano n. 370 del 2/10/1982 - Sped. in abb. post. gr. III n. 70 quale supplemento alla rivista Computer - Pubbl. inferiore al 70%

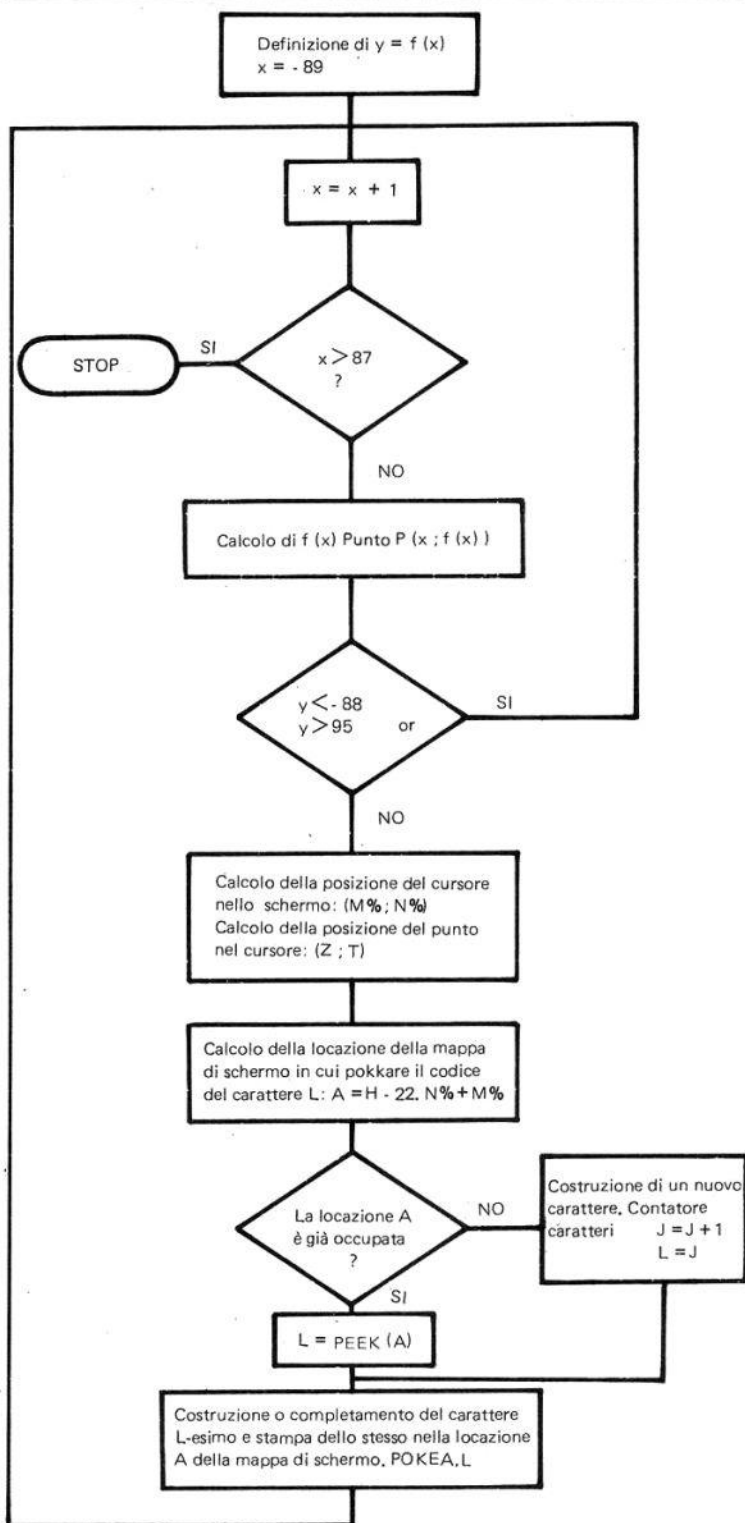
Grafici in alta risoluzione

Il BASIC del Vic-20, nella versione base, non possiede comandi grafici per l'alta risoluzione. Quindi, se si desiderano visualizzare istogrammi etc., bisogna ricorrere a manipolazioni (macchinipolazioni) piuttosto elaborate dei caratteri in dotazione, mentre è praticamente impossibile ottenere grafici di funzioni poichè la risoluzione che si ottiene gestendo il quarto di cursore per lo schermo del Vic è ancora troppo bassa (44 x 46).

LA MEMORIA di schermo del Vic 20 contiene 23 righe x 22 colonne per un totale di 506 caratteri o distinte posizioni del cursore. Questo è poi diviso in 8 x 8 = 64 punti. Quindi la massima risoluzione a cui può giungere il video è (22 x 8) x (23x8) = 176 x 184 = 32.384 punti. Non è molto ma sarebbe già sufficiente per rappresentare grafici di funzioni se il Basic del Vic possedesse alcune istruzioni grafiche elementari.

Questo programma permette di visualizzare in "alta risoluzione" il grafico di funzioni di una variabile senza far uso di nessuna espansione grafica o di memoria. Il programma è ridotto all'essenziale e lascia alcuni problemi aperti, ma chiunque avrà la pazienza di leggerne il commento e capire come funziona potrà modificarlo in base alle sue esigenze. Onde evitare eccessive spiegazioni, è doveroso ed opportuno citare almeno due riferimenti utili alla lettura del seguito:

a/ il capitolo 6 del libro "Impariamo a programmare in Basic con il Vic" della dr. R. Bonelli (Milano, 1981) dove si spiega, tra l'altro, come sono strutturati i 4K di ROM che contengono l'immagine dei caratteri e dove si trovano gli indirizzi che permettono di stabilire o leggere come è divisa la memoria utente;

b/ l'articolo "La definizione dei caratteri con il Vic" di C. Del Forno (BIT, luglio-agosto 1982), dove si mostra, con un program-

ma didatticamente molto chiaro, come sfruttare la possibilità del Vic di indirizzare i puntatori dei caratteri in una zona di RAM in cui possono esserne definiti di nuovi.

Il funzionamento del programma che segue, nelle linee essenziali, è questo: si calcola il valore della funzione y = f (x) per un insieme di valori crescenti di x (ad es. $-88 < x < 88$ step 1) e, in base ai risultati ottenuti, vengono automaticamente costruiti e stampati, un punto alla volta, dei caratteri, il cui susseguirsi sul video costituisce, approssimativamente, il grafico della funzione.

Il programma

• *Linee 1-20.* La poke 36869,254 permette di costruire una nuova serie di immagini dei caratteri a partire dal byte 6144. Sono quindi disponibili 7679-6143 = 1536 byte corrispondenti a 1536/8 = 192 caratteri. Questi 1536 byte vengono sottratti ai 3.5K di RAM utente ed è necessario che di ciò il Vic venga informato, soprattutto riguardo alla memorizzazione delle stringhe che viene effettuata per indirizzi decrescenti a partire, normalmente, dal fondo della memoria. A questo provvedono le pokes della linea 10. Inoltre vanno messi inizialmente a zero tutti i bytes utilizzati per la costruzione dei nuovi caratteri. (Vengono azzerati solo i byte dal 6144 al 6783 anzichè al 7679 per impiegare minor tempo e perchè 80 caratteri sono quasi sempre sufficienti per i grafici. Se così non fosse si può facilmente apportare una correzione). La poke 36879,104, infine, fissa il colore sfondo/schermo e può essere cambiata a piacere.

• *Linee 30-50.* Sono abbastanza chiare. La linea 45 fa in modo che la funzione non venga calcolata nei punti in cui non è definita evitando dei break per errore. Deve essere compilata con cura ogni volta che si introduce una nuova funzione. Per illustrare il significato dei parametri B, C, D, E che compaiono nella linea 50 e che vengono introdotti con i quattro input iniziali occorre far riferimento ad alcune nozioni di analisi matematica elementare.

Data una funzione y = f (x) ed un numero k si ha che:

1/ k. f (x) "dilata" l'unità di misura sull'asse y di un fattore k; in pratica la funzione si allunga o si accorcia verticalmente a seconda che k sia maggiore di uno oppure compresa tra zero ed uno;

2/ f (x/k) opera analogamente sull'asse x;

3/ y = f (x) + k sposta verticalmente la funzione di k unità verso l'alto o verso il basso a seconda che k sia positivo o negativo;

4/ y = f (x - k) sposta orizzontalmente la funzione di k unità verso destra o sinistra a seconda che

```
1 PRINTCHR$(147):PRINTCHR$(5):POKE36879,104
2 PRINT:PRINT"UNITA'ASSE X =";C:PRINT:INPUTC
3 PRINT:PRINT"UNITA'ASSE Y =";B:PRINT:INPUTB
4 PRINT:PRINT"ALTO-BASSO =";D:PRINT:INPUTD
5 PRINT:PRINT"DESTRA-SINISTRA =";E:PRINT:INPUTE:PRINTCHR$(147)
10 FORI=6143TO6783:POKEI,0:NEXT:POKE51,255:POKE52,24:POKE55,255:POKE56,24
20 POKE36869,254:H=7933:G=6144:X=-89
30 DEFFNF(X)=SIN(X)/X
40 X=X+1:IFX>87GOTO75
45 IFX/C-E=0GOTO40
50 Y=INT(B*FNF(X/C-E)+D):IFY>95ORY<-88GOTO40
55 M%=X/8:Z=X-M%*8:N%=Y/8:T=Y-N%*8
60 A=H-22*N%+M%:IFPEEK(A)<>32THENL=PEEK(A):GOTO70
65 J=J+1:L=J:IFJ=32GOTO65
70 K=G+(7-T)+8*L:POKEK,2↑(7-Z)+PEEK(K):POKEA,L:GOTO40
75 GETS$:IFS$=""GOTO75
80 POKE36869,240:J=0:GOTO1
```

Personal Computer professionali

CODICE	PRODOTTO	PREZZO LIRE (IVA esclusa)
	SERIE 4000 40 Colonne Video, Basic 4.0	
CBM 4016	**CPU 16K RAM**	1.750.000
CBM 4032	**CPU 32K RAM**	2.190.000
CBM 2031	**Floppy Disk** drive singolo 171K byte	1.065.000
CBM 4040	**Floppy Disk** doppio drive 343K byte	2.190.000
CBM 4022	**Stampante ad aghi** bidirezionale	1.095.000
	SERIE 8000 80 Colonne Video, BASIC 4.0	
CBM 8032-SK	**CPU 32K RAM** tastiera separabile, video orientabile	2.700.000
CBM 8096-SK	**CPU 96K RAM** tastiera separabile, video orientabile	3.350.000
CBM 9000	**Micro Main Frame Computer** 134K RAM a doppio CPU (6502-6809)	3.400.000
CBM 8050	**Floppy Disk** doppio drive 1M byte in linea	2.825.000
CBM 8250	**Floppy Disk** doppio drive a doppia faccia con 2M byte in linea	3.450.000
CBM 9060	**Hard Disk Winchester** con 5M byte in linea	4.950.000
CBM 9090	**Hard Disk Winchester** con 7.5M byte in linea	6.100.000
CBM 6400	**Stampante a margherita** Alta qualità di stampa, bidirezionale, 40 cps (utilizzabile anche con carta da bollo)	2.950.000
CBM 8023	**Stampante ad aghi** bidirezionale 150 cps - 132 colonne	1.855.000
	COMMODORE 64 40 colonne, colore, sintetizzatore di suono	
CBM 64	**CPU 64K RAM** con alta risoluzione grafica incorporata, 256 combinazioni di colore, collegabile ad un comune televisore Con interfaccia T-2 (IEEE 488) collegabile alle periferiche Commodore Serie 4000 e 8000	825.000
1541	**Unità a disco (Floppy Disk)** Veloce unità di memoria di massa al alta capacità. Può immagazzinare fino a 170.000 caratteri su ogni singolo disco	680.000
1525	**Unità stampante** Stampante ad aghi, matrice 5x7 caratteri grafici, 50 caratteri al secondo, 80 colonne.	550.000
1530	**Registratore a cassette** Per memorizzare facilmente programmi e dati su normali cassette magnetiche	120.000
T-2	**Interfaccia IEEE 488** Consente il collegamento di tutte le periferiche Commodore	175.000

continua ➧

Personal Computer professionali

seguito

CODICE	PRODOTTO	PREZZO LIRE (IVA esclusa)
	SOFTWARE STANDARD	
	Compilatori	
S-11	Compilatore «Pet Speed» (primo compilatore BASIC ottimizzato aumenta la velocità di esecuzione dei programmi di circa 40 volte)	595.000
	Linguaggi	
S-21	COBOL e aggiornamento per il CBM «MMF 9000»	95.000
S-22	TCL - PASCAL	325.000
S-23	UCSD - PASCAL	275.000
	Data Base manageriale	
S-31	«The Manager» per la serie 8000 (versione italiana)	620.000
	Planificazione manageriale (Spread Sheet)	
S-41	Visicalc 96 per il CBM 8096/8096SK	400.000
S-42	Visicalc per il CBM 4032/8032/8032SK	300.000
S-43	Calc-Result, il nuovo «Spread-Sheet»: — fino a 32 pagine disponibili — 4 parti di pagine differenti visualizzabili contemporaneamente — output grafico ed istogrammi — per la serie 8000	350.000
	NET-WORK **Offre la possibilità di collegare fino a 8 CPU (4032 - 8032) con stampante e floppy disk.**	
MBS-100	**Unità madre**	490.000
MBS-100-3	**Unità figlia** con 3 m di cavo	310.000
MBS-100-7	**Unità figlia** con 7 m di cavo	340.000
	SCHEDE AGGIUNTIVE	
B-1	**64K RAM** e nuovo sistema operativo «LOS-96» per incrementare fino a 96K RAM il CBM 8032	760.000
B-2	**CP/MAKER** incrementa la memoria interna di 64K RAM e permette l'uso di tutti i programmi CP.M 8 bit disponibili. Compatibile con serie 3000, 4000, 8000	1.450.000
B-3	**Scheda ad alta risoluzione grafica** con 32K RAM, 25 nuovi comandi BASIC e due modi di risoluzione: 1 pagina video con 512 per 512 punti 2 pagine video con 512 per 256 punti	720.000
	ALTRE PERIFERICHE	
CBM 8010	**Acoustic Coupler**, 300 Baud/sec	595.000
CBM 8075	**Plotter**	3.950.000
C-1	**Cavo P/I**	85.000
C-2	**Cavo I/I**	95.000
	DISCHI FLESSIBILI 5" 1/4 «COMMODORE»	
D-1	Scatola di 10 dischetti a singola faccia	75.000
D-2	Scatola di 10 dischetti a doppia faccia	115.000

Prezzi in vigore dal 31/3/83 e soggetti a modifica senza priore notifica

k sia positivo o negativo.

Agendo su questi quattro operatori si ha a disposizione una sorta di grandangolare che ci dà una visione generale (si fa per dire) della funzione, trasformabile in un teleobiettivo che ce ne mostra i dettagli e che può essere "puntato" su qualunque zona del piano cartesiano (compatibilmente con la capacità di calcolo del Vic).

• *Linee 55-65.* A questo punto il calcolatore ha le coordinate x e y di un punto della funzione e il problema è come fargli stampare il puntino (x;y) sullo schermo senza modificare ciò che in esso è già presente. Il procedimento è questo:

a/ si calcola quale delle 506 locazioni dello schermo "contiene" il punto (x;y). Ciascuna locazione può essere occupata da un carattere di 8x8 punti;

b/ si calcola quale dei 64 punti del carattere da stampare deve essere "acceso".

Nella linea 55, M% ed N% sono le coordinate del cursore nello schermo assumendo come origine (per comodità di calcolo) la locazione in basso a sinistra, mentre Z e T sono le coordinate del punto nel cursore.

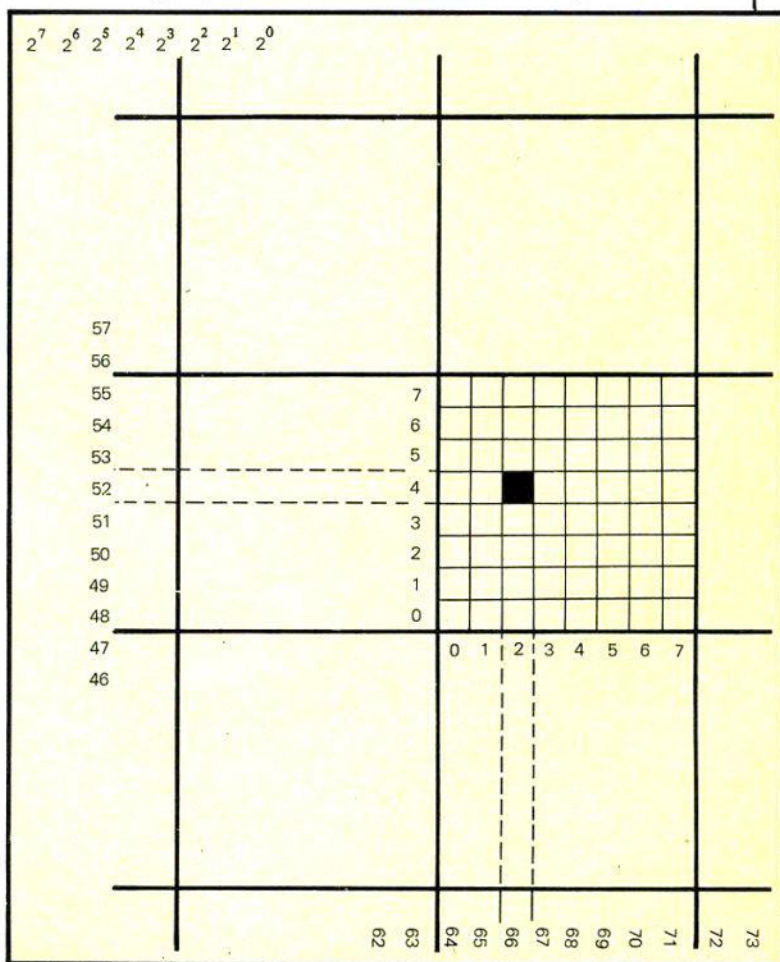
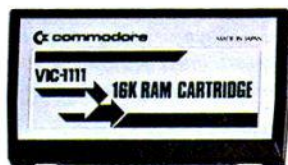
Conoscendo M% ed N% si può calcolare in quale locazione A della *video-RAM* si dovrà pokkare il codice del carattere (H è la locazione fissata, ai fini della visualizzazione, come origine degli assi).

Si possono ora verificare due casi: o la locazione A è occupata, e quindi contiene un codice diverso dal 32 che è lo space, oppure contiene il 32. Nel primo caso occorre conoscere il codice del carattere che la occupa poichè proprio quello deve essere modificato. Nel secondo caso occorre definire un nuovo carattere da associare a quella locazione. Si incrementa dunque un contatore J dei caratteri di una unità. L, codice del carattere, è in tal caso uguale a J.

• *Linea 70.* Il passo decisivo con-

Fig. 3

In questo esempio le coordinate del punto da stampare sono (66;52). Le coordinate del cursore nello schermo sono i quozienti interi delle divisioni: M% = 66/8 = 8 e N% = 52/8 = 6. Le coordinate del punto nel cursore sono i resti delle precedenti divisioni e cioè Z = 2 e T = 4.

Occorre poi calcolare a quale locazione della video-RAM corrispondono le coordinate x ed y del cursore. A tale scopo è sufficiente ricordare che, fissata una locazione H come origine, per spostarsi di y righe ed x colonne si deve calcolare H + 22.y + x.

un vero sistema

Cassette per espansione di memoria.
VIC 20 ha una memoria di base di 5 K bytes. Per aumentarla sino a 32 K bytes hai a disposizione tre cassette da 3, 8 e 16 K RAM. Basta inserirle nel VIC oppure nell'apposito Memory expansion board.

VIC 20 è un sistema interamente espandibile grazie alla semplice aggiunta di moduli, questi che vedi e molti altri.

VIC 20 Computer.
Questo è il favoloso VIC 20 computer. 5 K-Bytes espandibili a 32. 24 colori. Note musicali. Collegabile all'unità nastro, all'unità disco e alla stampante. Collegabile in Modem con le normali linee telefoniche.

VIC 20 Single drive floppy disk. Unità a disco.
Questo "floppy" consente di sfruttare tutta la potenzialitá del VIC 20. Mette a disposizione un metodo veloce ed efficiente per la memorizzazione e il recupero dei dati e dei programmi.
Abbinando il floppy disk alla stampante, il VIC 20 diventa un computer system ideale per il piccolo imprenditore, per il professionista, per le ricerche.

Cassette Unit. Unità a nastro.
È questo il primo degli accessori del VIC. Viene utilizzata per memorizzare dati e programmi o per inserire cassette con programmi già pronti.
Si collega direttamente con il VIC 20.

Stampante VIC 20.
Come ogni periferica VIC 20 questa stampante è prodotta con alta specializzazione e basso costo.
Ha mille utilizzi.
Per fare copie dei programmi, per fare lettere circolari, grafici, dati di lavoro, etc. Per realizzare programmi di word processing.

siste nel calcolo dell'indirizzo del byte da modificare e del numero (decimale) da pokkarvi per mettere ad 1 (uno) un suo determinato bit lasciando invariato tutto il resto. Il byte è il K-esimo e la formula per calcolarlo è riportata nel listato in forma piuttosto trasparente.

Per quel che riguarda il numero da pokkare nel byte K è sufficiente ricordare che per porre ad 1 l'n-esimo bit (supposto, come in questo caso, che sia a O) di un byte bisogna pokkare, in decimale, il numero 2^n oppure 2^{7-n} se i bit vengono numerati da sinistra verso destra.

Finalmente, compiute tutte queste operazioni, il Vic completa l'esecuzione della linea 70 e, mediante la Poke A, L, un nuovo puntino si aggiunge al grafico che a poco a poco compare sullo schermo.

Terminato il grafico, attraverso un GET, il controllo passa all'utente il quale potrà fissare nuove unità di misura sugli assi o spostare l'origine degli stessi.

Se si desidera introdurre una nuova funzione bisogna premere i tasti RUN-STOP e RESTORE e definirla nella linea 30. Se ci sono alcuni punti o intervalli in cui non è definita occorre compilare opportunatamente la linea 45.

Fig. 4

Supponiamo, come mostra la figura, di conoscere le coordinate di un punto (Z;T) = (2;4) che deve essere "acceso" nel carattere L–esimo. Dobbiamo innanzitutto calcolare l'indirizzo del byte da modificare e il numero da pokkarvi per mettere ad 1 un suo determinato bit. Se il codice del carattere è L allora il suo byte 0 si trova alla locazione 6144 + L.8. Come risulta chiaramente dalla figura, l'indirizzo del byte è K = 6144 + L.8 + (7 - 4) e il numero da pokkarvi è $2^{(7/2)}$. Se inoltre vogliamo lasciare invariata la situazione degli altri bit del byte (che nel nostro caso è il byte 3) dobbiamo fare in modo che in esso venga pokkato il numero $2^{(7/2)}$ + Peek (K). Siamo sicuri che non ci saranno delle interruzioni per illegal quantity error poichè inizialmente i bytes utilizzati per la costruzione dei caratteri sono stati posti tutti a zero e la variabile x aumenta di una unità alla volta, quindi non accadrà mai di dover mettere ad 1 più di una volta lo stesso bit dello stesso byte.

Conclusioni

Il programma gira "abbastanza velocemente" e i grafici che si ottengono danno un'idea attendibile dell'andamento della funzione nell'intervallo impostato. E' necessario però assegnare in modo opportuno, procedendo eventualmente per tentativi, le unità di misura sugli assi.

Le cause che possono far fermare l'esecuzione del programma sono, in genere, due:

a/ 80 caratteri non sono sufficienti; in tal caso si può correggere la linea 10

10 FORI = 6144T07679:POKEI, 0:NEXT:...etc.

b/ Non è stata compilata bene la linea 45. Ad es. se la funzione non è definita nei punti x = -3, x = 5,...etc. la linea 45 diventa:

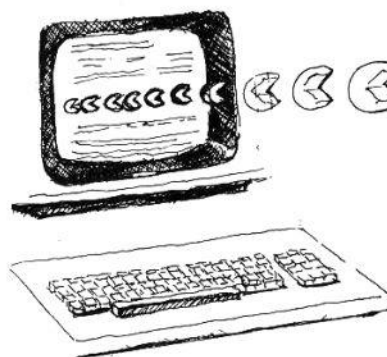
4 5 IF X/C-E = -30RX/C-E = 50RX/C-E =...GOTO40

Analogamente si procede se la funzione non è definita in uno o più intervalli. In ogni caso se si desidera conoscere la causa del break si deve battere POKE36869,240.

Il programma è suscettibile di alcuni utili miglioramenti come:
- visualizzare gli assi cartesiani ed eventualmente i grafici di più di una funzione alla volta;
- migliorare la grafica delle linee e pendenza elevata;
- lavorare in coordinate polari.

Sono disponibili, a tale scopo, circa 500 bytes e, volendo, la mappa dei nuovi caratteri può essere spostata "in avanti" di circa 200 bvtes. ■

di **Gianfranco Bo***

* Corso Risorgimento, 108/3
16040 S. Salvatore (Genova)

VIC SCROLL

Una routine inventa-giochi per lo scroll laterale del Vic.

AVRETE notato che quando il cursore lascia l'ultima riga dello schermo per iniziarne un'altra, tutte le righe superiori salgono, lasciando il posto per una nuova riga bianca.

In pratica la SCROLL ROUTINE, richiamata dal Sistema Operativo, ricopia ogni riga in quella immediatamente superiore, come si può capire meglio usando un piccolo artificio: scrivete nell'ultima riga in basso: POKE 37157, ∅ premete il RETURN e tenete quindi premuto il CRSR DOWN. Vedrete, allora, l'effetto della SCROLL ROUTINE alquanto rallentato.

Dopo aver resettato il Vic proviamo a scrivere una routine in BASIC che abbia lo stesso effetto: (vedi LISTING 1). Dopo il RUN vedrete che lo scopo è raggiunto, anche se in modo estremamente lento: la velocità non è certo la miglior caratteristica del BASIC.

La SCROLL ROUTINE fa parte del Sistema Operativo (Kernel) ed inizia alla locazione decimale 59765; per richiamarla basta dunque scrivere SYS 59765 e premere il RETURN. Proviamo ad usarla con un programmino (vedi LISTING 2).

Dopo il RUN vedrete una specie di nevicata. Ora anche i più distratti di voi si saranno accorti che questa neve è per lo meno "strana": infatti sembra *SALIRE* mentre è risaputo che nella realtà essa *SCENDE*. Ed eccoci al nostro problema: perchè lo SCROLL va solo verso l'alto e non anche verso il basso, a sinistra o a destra come vorremmo noi?

La maggior parte dei giochi di movimento (es. corse d'auto), hanno infatti un innaturale movimento verso il basso (parlo dei giochi fatti in casa s'intende, quelli inventati da noi). Ed allora, visto il problema, perchè non inventare delle routines che permettano uno SCROLL verso il basso (DOWN SCROLL), a sinistra (LEFT SCROLL) ed a destra (RIGHT SCROLL)?

Detto fatto, prima o poi infatti scoprirete che ben poche sono le cose che *NON* si possono fare con il Vic.

Ritornando alla nostra routine in BASIC del LISTING 1 che simulava lo SCROLL verso l'alto, proviamo ora a riscriverla in modo che questa volta lo SCROLL sia verso il basso: (vedi LISTING 3). Lo scopo è raggiunto anche questa volta, solo che, essendo troppo lenta, la routine non ci può essere di alcuna utilità. Ora tutti sanno che la caratteristica principale dei programmi scritti in linguaggio macchina è la velocità, non ci resta quindi altro da fare che tradurre suddetto programma in linguaggio macchina ed il gioco è fatto. Allo stesso

Da AAE a ZOOMING 7065 VOCI NEL NUOVO DIZIONARIO DELL'INFORMATICA

UN DIZIONARIO COSI' SOLO COMPUTER POTEVA FARLO

Made in Italy

Il primo grande dizionario d'informatica non tradotto dall'inglese, del quale un grande editore americano come Auerbach abbia richiesto i diritti d'autore per il mercato statunitense, finalmente è disponibile. Sulla rivista la pubblicazione delle dispense proseguirà fino alla Z, ma chi vuole disporre fin d'ora dell'opera completa, rilegata in cartone telato, può richiederla alla redazione e profittare dell'offerta di lancio.

Migliaia di voci in inglese ed in italiano con le definizioni dei principali organismi di standardizzazione internazionali (IEEE, ISO, ANSI, ecc.) fanno di questo volume un indispensabile strumento di lavoro per quanti operano professionalmente nell'edp.

Anni di lavoro dei tre coautori e dell'intera redazione di "Computer" testimoniamo un impegno che nessun altro editore italiano, grande medio o piccolo, aveva mai tentato.

Aggiornamento telefonico personalizzato

E, come se non bastasse, la redazione di Computer s'impegna a rispondere singolarmente a tutte le richieste degli acquirenti del dizionario per tutte le voci nuove che non dovessero figurare in questa prima edizione ed a pubblicarne un aggiornamento periodico.

Spettabile redazione,
desidero ricevere il "Dizionario dell'informatica" da voi edito e profittare della speciale offerta di lancio

☐ *Una copia del dizionario a lire 84.000*
☐ *Una copia del dizionario + l'abbonamento a Computer per un anno, a partire dalla data dell'ordine, al prezzo complessivo di lire 94.000*
☐ *Una copia del dizionario + l'abbonamento per un anno a Computer ed a Computer Club per complessive lire 100.000*
☐ *Ho già rinnovato l'abbonamento a Computer, pertanto dedurrete l'importo da me pagato dal prezzo cumulativo dizionario + abbonamento di lire 94.000.*

☐ A tale scopo invio in allegato assegno n° .... Banca .... per l'importo di lire ....
☐ Ho versato oggi stesso l'importo di lire .... sul vostro c/c postale n. 11909207
☐ Pagherò a ricevimento della fattura, che vi prego intestare come segue:

Ditta/o ragione sociale: ....
Via .... n° .... CAP ....
Città .... Partita IVA ....

Il mio indirizzo è il seguente: Nome .... Azienda ....
Qualifica .... N° .... CAP .... Città ....
Via ....
Tel. .... F Firma ....

modo si possono creare le routines per lo SCROLL a sinistra e per quello a destra.

Il risultato è il Vic SCROLL (vedi LISTING 4). Esso è scritto in BASIC, naturalmente, ed è concepito in modo che le tre routines in linguaggio macchina (DOWN SCROLL, LEFT SCROLL e RIGHT SCROLL), vadano a posizionarsi automaticamente nella parte più alta della RAM a vostra disposizione, che diminuisce così di 340 BYTES, tanti quanti ne occupano le tre routines. Il programma funziona con qualsiasi configurazione di memoria del Vic ed alla fine si autodistrugge (il programma BASIC non il VIC, vedi il NEW alla fine della riga 7) dandovi i valori decimali di accesso alle tre routines. Il valore di accesso alla UP SCROLL è sempre la stessa ed è quella del KERNAL.

L'uso di queste routines dipende solo dalla fantasia; provate a sostituire al SYS 59765 nella riga 4 del LISTING 2 il valore datovi da programma per il DOWN SCROLL (SYS 7343 per il Vic senza espansioni) e vedrete che la vostra neve diventerà più naturale.

Al LISTING 5 troverete un programmino che fa uso delle tre routines e che, sebbene sia ai limiti della demenzialità, spero vi serva come spunto per le vostre invenzioni: esso prevede l'uso del joystick ed è formulato per la versione base del Vic. Se avete dotato il vostro VIC di espansioni di memoria, cambiate i valori di D, L ed R alla riga Ø con quelli fornitivi dal Vic SCROLL rispettivamente per il DOWN, LEFT e RIGHT SCROLL. Inoltre, se la memoria aggiunta supera i 3K, cambiate i valori 7680, 38400, 7933 e 38653 alle righe 1, 2 e 3 rispettivamente con 4096, 37888, 4349 e 38141 semplice no?

Ed ora alcune precisazioni sul Vic SCROLL:

1/ Subito dopo aver ricopiato il programma salvatelo su cassetta: un eventuale errore di copiatura porterebbe ad un inevitabile CRASH con relativa perdita del programma.

2/ Le linee 14, 18 e 26 non hanno bisogno di REM dato che non verranno mai valutate dall'interprete BASIC.

3/ Dopo aver premuto lo SHIFT (come vi richiederà il programma), sarete liberi di fare ciò che vorrete e cioè: o di scrivere un programma ex novo oppure di caricarne uno dalla cassetta come siete soliti fare: le tre routines si manterranno al di fuori (o meglio, al di sopra) di ciò che accade nella RAM, essendo protette dai puntatori di fine memoria. Ad esse vi potrete accedere con i comandi che il programma vi avrà indicato e di cui avrete preso nota.

4/ Non LISTate il programma prima della sua esecuzione. Ne otterreste un blocco oppure un incredibile SYNTAX ERROR IN 7. Fatelo partire subito dopo averlo caricato dalla cassetta, oppure caricatelo usando SHIFT & RUN/STOP (prima di caricarlo inizializzate il Vic con SYS 64802 oppure spegnetelo e riaccendetelo).

LISTatelo solo se lo dovete correggere e poi risalvatelo sulla cassetta. (Tutto quanto detto in questo ultimo punto vale solo se avete il Vic senza espansioni di memoria oppure con l'espansione 3K).

E, per concludere, una domanda agli appassionati: "Che funzione ha la variabile F? ". ▪

di **Filippo Pozzi**
Via XX Settembre, 80/B - 27058 Voghera (PV) - Tel. 0383/5664090

Listing 1

```
0 FORR=0TO505-22
1 POKE7680+R,PEEK(7680+22+R)
2 POKE38400+R,PEEK(38400+22+R)
3 NEXT
4 FORR=0TO21
5 POKE8164+R,32
6 NEXT:GOTO0
```

Listing 2

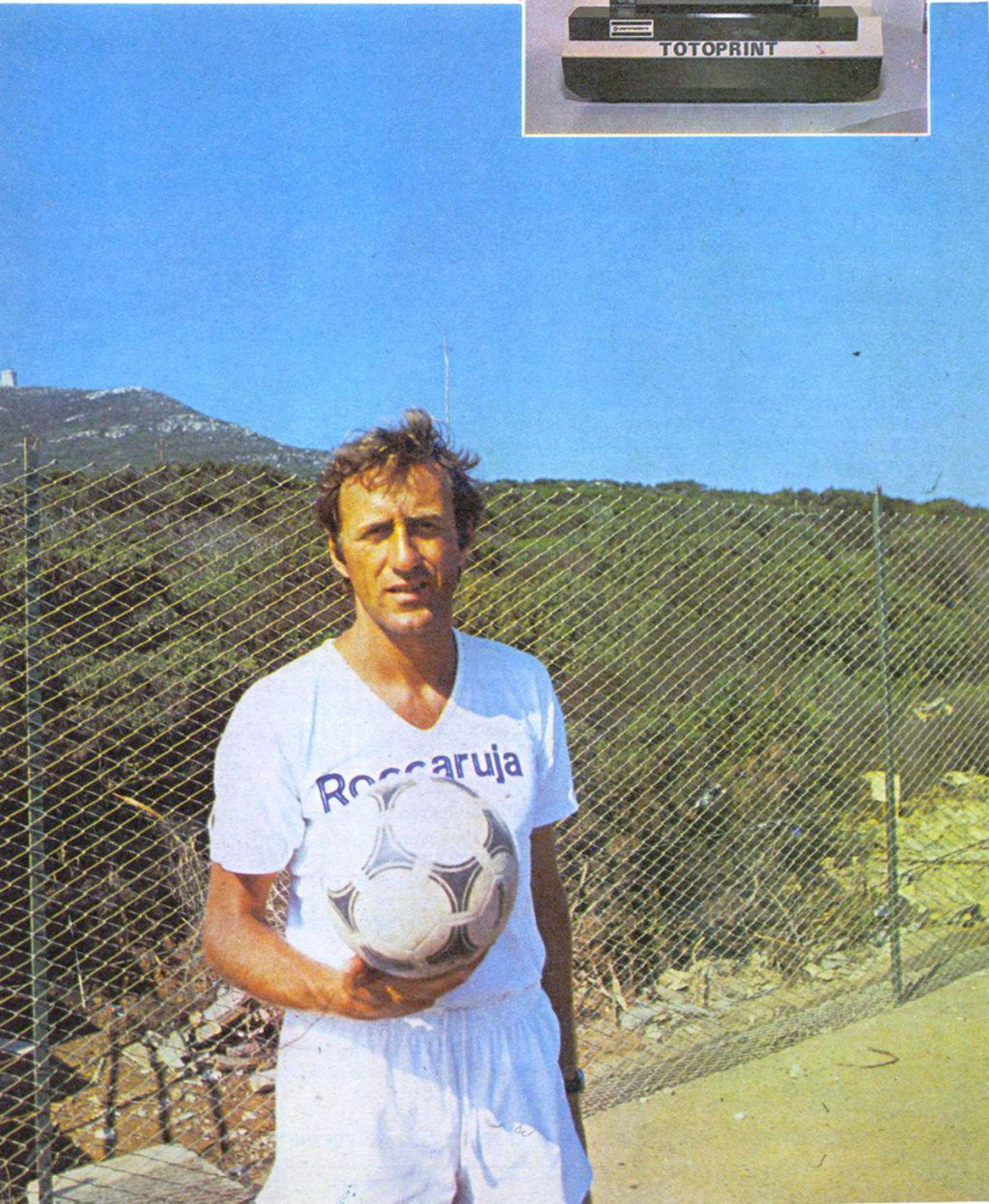
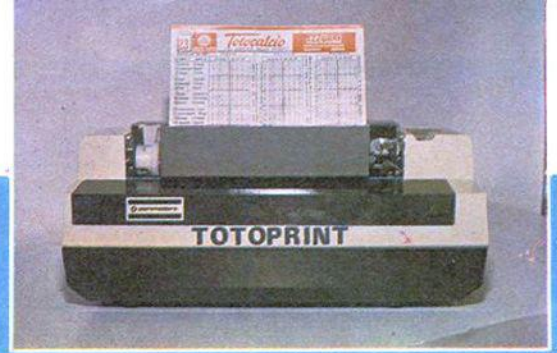
```
0 POKE36879,110
1 A=RND(1)*505
2 POKE7680+A,46
3 POKE38400+A,1
4 SYS59765:GOTO1
```

Listing 3

```
0 FORR=505-22TO0STEP-1
1 POKE7680+22+R,PEEK(7680+R)
2 POKE38400+22+R,PEEK(38400+R)
3 NEXT
4 FORR=0TO21
5 POKE7680+R,32
6 NEXT:GOTO0
```

```
0 REM-------VIC-SCROLL-------BY-F.POZZI---
1 F=PEEK(260):A=PEEK(55):B=PEEK(56):IFA<81THENPOKE55,A+174:POKE56,B-2:GOTO3
2 POKE55,A-81:POKE56,B-1
3 K=PEEK(55)+256*PEEK(56):PRINT"▒▒▒SCREEN SCROLL ROUTINES"
4 PRINT"▒▒▒UP SCROLL....SYS 59765":PRINT"▒DOWN SCROLL..SYS"K+1
5 PRINT"▒LEFT SCROLL..SYS"K+52:PRINT"▒RIGHT SCROLL.SYS"K+195
6 FORR=K+1TOK+337:READA:GOSUB8:POKER,A:NEXT:PRINT"▒▒▒ PRESS ▒ SHIFT ▒ TO END▒"
7 WAIT653,1:FORR=0TO21:SYSK+52:NEXT:NEW
8 IFFTHENRETURN
9 IFA=16THENA=30
10 IFA=17THENA=31
11 IFA=148THENA=150
12 IFA=149THENA=151
13 RETURN
14 DOWN SCROLL ROUTINE--------------------
15 DATA162,230,189,253,16,157,19,17,189,253,148,157,19,149,202,224
16 DATA255,208,239,162,252,189,0,16,157,22,16,189,0,148,157,22,148
17 DATA202,224,255,208,239,162,0,169,32,157,0,16,232,228,23,208,248,96
18 LEFT SCROLL ROUTINE--------------------
19 DATA169,0,133,1,133,251,169,16,133,2,169,148,133,252,160,0,200,177,1,136,145
20 DATA1,200,177,251,136,145,251,200,192,21,208,239,169,32,145,1,24,169,22,101
21 DATA1,133,1,24,169,22,101,251,133,251,169,242,197,1,208,213,162,0,232,189,242
22 DATA16,202,157,242,16,232,189,242,148,202,157,242,148,232,224,21,208,235,169
23 DATA32,157,242,16,169,8,133,1,133,251,169,17,133,2,169,149,133,252,160,0,200
24 DATA177,1,136,145,1,200,177,251,136,145,251,200,192,21,208,239,169,32,145,1
25 DATA24,169,22,101,1,133,1,24,169,22,101,251,133,251,169,250,197,1,208,213,96
26 RIGHT SCROLL ROUTINE--------------------
27 DATA169,0,133,1,133,251,169,16,133,2,169,148,133,252,160,21,136,177,1,200,145
28 DATA1,136,177,251,200,145,251,136,192,0,208,239,169,32,145,1,24,169,22,101,1
29 DATA133,1,24,169,22,101,251,133,251,169,242,197,1,208,213,162,21,202,189,242
30 DATA16,232,157,242,16,202,189,242,148,232,157,242,148,202,224,0,208,235,169
31 DATA32,157,242,16,169,8,133,1,133,251,169,17,133,2,169,149,133,252,160,21,136
32 DATA177,1,200,145,1,136,177,251,200,145,251,136,192,0,208,239,169,32,145,1
33 DATA24,169,22,101,1,133,1,24,169,22,101,251,133,251,169,250,197,1,208,213,96
```

LISTING #5

```
0 POKE37154,127:POKE36877,240:POKE161,0:D=7343:U=59765:L=7394:R=7537:PRINT"▒"
1 A=RND(1)*505:B=RND(1)*8:C=PEEK(37137)+PEEK(37152):POKE7680+A,81:POKE38400+A,B
2 IFPEEK(7933)=81THENP=P+1:FORV=15TO0STEP-1:POKE36878,V:NEXT
3 POKE7933,87:POKE38653,2:IFPEEK(161)AND8THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒PUNTI"P:GOTO13
4 IFC=369THENSYSD:GOTO1
5 IFC=353THENSYSD:SYSR:GOTO1
6 IFC=241THENSYSD:SYSL:GOTO1
7 IFC=357THENSYSR:GOTO1
8 IFC=349THENSYSU:SYSR:GOTO1
9 IFC=237THENSYSU:SYSL:GOTO1
10 IFC=365THENSYSU:GOTO1
11 IFC=245THENSYSL
12 GOTO1
13 PRINT"▒▒▒▒FIRE▒   PER  CONTINUARE":WAIT37137,32,32:FORS=0TO21:SYSL:NEXT:RUN
```

Nella foto grande il "Commodoriano" Mauro Bellugi. Nel riquadro una riproduzione del TOTPRINT, per la stampa di schedine multiple di cui si parla in questo articolo.

Come fare "13" col Vic

NON vi è nulla di strano nel fatto che si sia pensato di portare nel "computer" una disciplina del tutto nuova ed affascinante come quella relativa al Totocalcio.

Gli esperti del settore amano definirla "sistemistica", con ciò volendo alludere ad una vera e propria scienza, che ha come obiettivo la realizzazione di metodi estremamente rigorosi che consentano di accedere al sogno (comune a molti, e solo da pochi realizzato) di una grossa vincita che risolva una buona dose dei problemi del quotidiano vivere.

In fondo ciascuno di noi, davanti a un computer, avrà prima o poi pensato a formulare la propria "schedina" in base a criteri di un qualche tipo e di una qualche validità. Come, in fondo, è risaputo che esistono in commercio onesti programmini relativi all'argomento; ed è altrettanto noto che chiunque disponga di un video ed una tastiera nel proprio ufficio, ha, prima o poi, proposto ai colleghi di tirar fuori le diecimila lire necessarie per completare una giocata "fresca di tabulato". Non si è mai avuto notizia di persone che abbiano goduto di grossi ed effettivi vantaggi dall'uso di queste tecniche, e ciò quasi sempre per una ragione piuttosto importante: non basta essere un conoscitore di elettronica o programmazione per realizzare dei sistemi realmente validi. Bisogna, per così dire, essere del mestiere, intendersi di metodologie di gioco molto sottili e collaudate, avere avuto realmente un'esperienza "sul campo", se è vero che, quasi sempre, sistemi che sembrano eccellenti alla vigilia (per l'occhio profano del dilettante) risultano poi delle vere e proprie "frane" e viceversa.

Il fatto nuovo è che, questa volta, a proporre dei programmi (e, soprattutto, degli strumenti) per il Totocalcio è un professionista in materia, il dott. Vincenzo Carchidi, collaboratore tecnico del settimanale specializzato Totocorriere ed autore e curatore della rubrica dall'eloquente titolo "Dal Computer al Sistema".

L'altro fatto importante è costituito dalla scelta di uno strumento economico, brillante e largamente diffuso come il Vic 20 per l'elaborazione di sistemi e calcoli complessi finora (e solo occasionalmente) delegati quasi per intero a grossi e costosi computer.

Attraverso l'adozione di un interessante algoritmo, in effetti, i programmi ideati da Carchidi, consentono l'elaborazione di qualsiasi tipo di sistema, senza richiedere grandi impieghi di memoria, esigere unità a disco, o limitare la fantasia e l'inventiva dell'utente.

Il Vic 20, in pratica, porta in tutte le case la possibilità di una concezione professionale e scientifica del Totocalcio, in quanto, fra l'altro, oltre al programma-base, sono in cantiere progettazioni software per sistemi ridotti, elaborazioni scientifiche dei pronostici ed altre originalissime ed inedite applicazioni su cui l'autore ha voluto mantenere un rigido ed intransigente silenzio. "Che ci copino il più tardi possibile," ha commentato distaccatamente l'autore di questi programmi, con una punta di sarcasmo nei confronti di chi "ha fiutato l'affare" e sta con le orecchie tese a cogliere e copiare le novità.

E le novità, questa volta sono grosse davvero. L'autore del programma, in effetti, risulta oltretutto proprietario di un brevetto che si accinge a diventare internazionale: un alimentatore automatico di fogli singoli denominato "Totoprint", utilizzabile, appunto (fra le tante cose), per ottenere la stampa diretta ed automatica delle schedine.

La portata di questa applicazione assume una notevole importanza, laddove si pensi che esistono migliaia di ricevitorie, società di gioco e addirittura giocatori singoli che, settimanalmente, si scontrano con l'irrisolvibile (sinora, per lo meno) problema dell'errore e, soprattutto, della trascrizione manuale delle colonne.

Chi non ha mai provato a ricopiare 10.000 o addorittura 50.000 colonne a mano (il Coni non prevede infatti l'adozione di tabulati stampabili con elaboratori), non può lontanamente

```
5 REM                                    TOT9
10 REM
1000 DIMM$(13,3),C(3),D(3),N(3),P(13,5),BU$(13,64),BS$(80),BL%(20),Q$(13,3),R(13
,2)
1025 GS=0:E1=1
1100 INPUT"NUMERO RIGHE= ";NR
1150 IFNR<1ORNR>13THEN1100
1250 PRINT:PRINT
1300 PRINT"SCRIVERE IL PROSPET- "
1350 PRINT "TO UNA RIGA PER VOLTA"
1400 PRINT"SEPARANDO GLI ELEMEN-"
1450 PRINT"TI CON UNA VIRGOLA"
1700 PRINT:PRINT
1800 FORI=1TONR
1850 PRINT"RIGA NUMERO";I:INPUT" ";M$(I,1),M$(I,2),M$(I,3)
1900 IFM$(I,1)=""ANDM$(I,2)=""ANDM$(I,3)=""THEN1850
1950 FORK=1TO3:D(K)=0:NEXTK
2000 FORK=1TO3
2050 IFLEN(M$(I,K))=0THEN2400
2100 FORJ=1TOLEN(M$(I,K))
2150 IFMID$(M$(I,K),J,1)="1"THEND(1)=D(1)+1:GOTO2350
2200 IFMID$(M$(I,K),J,1)="2"THEND(2)=D(2)+1:GOTO2350
2250 IFMID$(M$(I,K),J,1)="X"THEND(3)=D(3)+1:GOTO2350
2300 GOTO1850
2350 NEXTJ
2400 NEXTK
2450 IFD(1)>1ORD(2)>1ORD(3)>1THEN1850
2500 NEXTI
2550 PRINT:PRINT
2600 FORI=1TONR
2650 FORK=1TO2:BL%(K)=LEN(M$(I,K)):NEXTK
2700 PRINTSPC(4);M$(I,1);SPC(4-BL%(1));M$(I,2);SPC(4-BL%(2));M$(I,3)
2750 NEXTI
2800 PRINT:PRINT:PRINT
2850 INPUT"VA BENE ";B$
2875 IFB$<>"SI"ANDB$<>"NO"THEN2850
2900 IFB$= "NO"THEN1250
2910 FORA=1TONR
2920 FORK=1TO3
2925 IFLEN(M$(A,K))>1THENE1=0
2930 NEXTK
2940 NEXTA
3000 PRINT"FORMULA":INPUT"DERIVATA";C(1),C(2),C(3)
3050 IFNR<>(C(1)+C(2)+C(3))THEN3000
3100 FORK=1TO3:N(K)=0:NEXTK
3150 FORK=1TO3
3200 FORI=1TONR
3250 IF(M$(I,K)<>"")THENN(K)=N(K)+1
3300 NEXTI
3350 NEXTK
3400 FORK=1TO3
3450 IFC(K)>N(K)THEN3000
3500 NEXTK
3600 FORK=1TO3:D(K)=0:NEXTK
3700 INPUT"SOLO NUMERO COLONNE ";SN$
3750 IFSN$<>"SI"ANDSN$<>"NO"THEN3700
3820 ST$="":GR$=""
3850 IFSN$="SI"THEN4150
3900 INPUT"SU STAMPANTE ";ST$
```

Commodore è vicina.

Napoli Parco S. Paolo

Commodore aggiunge un nuovo punto vendita e assistenza ai quasi 500 già operanti in Italia: CM Computermarket, Napoli Parco S. Paolo, Is. 9 Tel. 081/76.72.222. Qui trovi gente che la sa lunga sui computer e sulle qualità dei vari computer; professionisti che ti propongono sistemi con il miglior rapporto prezzo-prestazioni e una vasta gamma di soluzioni: gamma alla quale nessun produttore di "personal" si può oggi avvicinare. Anche per questo Commodore è prima in Italia. E in Europa. Ti aspettiamo.

ETHOS PISA

Buon prezzo

Commodore computer è conosciuta in tutto il mondo per l'eccezionale rapporto prezzo-prestazioni dei suoi sistemi. Il più interessante.

Anche per questo Commodore è fra le prime tre aziende mondiali di "personal" ed è prima in Europa. E prima in Italia.

Commodore Computer Italia dispone oggi di una rete che sfiora i cinquecento punti vendita.

non mente.

Con gente preparata e programmi personalizzati, pronti per risolvere i tuoi problemi.
Con Commodore sei in buone mani.
Commodore Italia s.r.l. - tel. 02/6125651

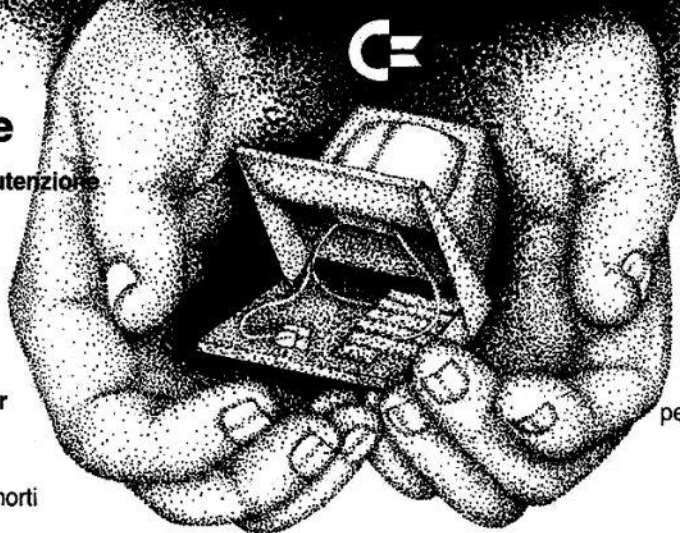
commodore
COMPUTER

ETHOS PISA

```
3950 IFST$<>"SI"ANDST$<>"NO"THEN3900
3970 IFE1=1THEN4150
3980 PRINT"A GRUPPI DI SISTEMI ":INPUT"INTEGRALI  ";GR$
3990 IFGR$<>"SI"ANDGR$<>"NO"THEN3980
4150 FORI=1TONR
4200 K=1
4250 FORJ=1TO3
4300 IFM$(I,J)=""THEN4400
4350 P(I,K)=J:P(I,4)=K:K=K+1
4400 NEXTJ
4450 NEXTI
4650 IFE1=0THEN5500
4700 CT=0:KP=1
4730 FORK=1TO3
4770 IFN(K)>0ANDN(K)<NRTHENCT=CT+1:KP=K
4800 NEXTK
4850 IFCT>1THEN5500
4900 IFKP=1THENKA=2:GOTO5050
4950 KA=1
5050 NB=N(KP):KB=C(KP):GOSUB63150:NC=CB
5100 NB=NR-C(KP):KB=C(KA):GOSUB63150:NC=NC*CB
5150 GOTO9100
5500 CT=0
5530 FORK=1TO3
5570 IFN(K)<>0THENCT=CT+1:KP=K
5600 NEXTK
5650 IFCT<>1THEN5950
5750 NC=1
5800 FORA=1TONR:NC=NC*LEN(M$(A,KP)):NEXTA
5850 GOTO9100
5950 IF(N(1)<>NR)OR(N(2)<>NR)OR(N(3)<>0)THEN9000
5970 FORA=1TONR
6030 IFLEN(M$(A,1))<>1THEN9000
6050 NEXTA
6130 ND=0
6150 FORA=1TONR
6170 IFLEN(M$(A,2))=2THENND=ND+1
6200 NEXTA
6250 MN=ND:IFMN>C(2)THENMN=C(2)
6300 NC=0
6350 FORK=0TOMN
6400 NB=ND:KB=K:GOSUB63150:RS=CB
6450 NB=NR-K:KB=C(2)-K:GOSUB63150:RS=RS*CB
6500 NC=NC+RS
6550 NEXTK
6600 GOTO9100
9000 PRINT" ":PRINT"   ATTENDERE   ":PRINT" "
9050 GOTO9200
9100 PRINT:PRINT"    TOTALE":PRINT"    COLONNE=";NC:PRINT
9150 IFSN$="SI"THEN13600
9200 IFSN$="SI"THEN10150
9250 IFST$="NO"THEN10150
9300 OPEN10,4
9350 IFGS<>0THEN9675
9375 S$="":FORI=1TO34:S$=S$+" ":NEXTI:S1$=S$+" PROSPETTO":PRINT#10,S1$;CHR$(13)
9400 FORI=1TONR
9450 FORK=1TO3:BS$(K)=M$(I,K):BL%(K)=LEN(M$(I,K)):NEXTK
9500 PRINT#10,S$;BS$(1);SPC(5-BL%(1));BS$(2);SPC(5-BL%(2));BS$(3)
9550 NEXTI
9600 PRINT#10," ":PRINT#10," ":PRINT#10," "
```

```
9650 GS=1
9675 FORK=1TO3:BS$(K)=STR$(C(K)):NEXTK
9700 PRINT#10,"FORMULA DERIVATA =";SPC(15);BS$(1);SPC(3);BS$(2);SPC(3);BS$(3);CH
R$(13)
9750 CLOSE10
10150 NC=0
10200 I=1
10250 J=1
10300 P(I,5)=J
10350 D(P(I,J))=D(P(I,J))+1
10400 IFD(P(I,J))<=C(P(I,J))THEN10550
10450 D(P(I,J))=D(P(I,J))-1
10500 GOTO12650
10550 I=I+1
10600 IFI<=NRTHEN10250
10625 IFGR$="SI"THEN40040
10800 FORA=1TONR
10850 L=P(A,5)
10900 R(A,1)=LEN(M$(A,P(A,L)))
10950 FORK=1TOR(A,1):Q$(A,K)=MID$(M$(A,P(A,L)),K,1):NEXTK
11000 NEXTA
11150 I1=1
11200 J1=1
11250 R(I1,2)=J1
11300 I1=I1+1
11350 IFI1<=NRTHEN11200
11400 NC=NC+1
11450 IFSN$="NO"ANDST$="NO"THEN11500
11475 IFNC=INT(NC/16)*16THENPRINT" C. TROVATE=";NC
11500 IFSN$="SI"THEN12200
11550 IFST$="SI"THEN12000
11600 F=NC-INT(NC/16)*16:IFF=0THENF=16
11650 FORA=1TONR:L=R(A,2):BU$(A,F)=Q$(A,L):NEXTA
11700 IFF<>16THEN11800
11750 GOSUB61050
11800 GOTO12200
12000 F=NC-INT(NC/64)*64:IFF=0THENF=64
12050 FORA=1TONR:L=R(A,2):BU$(A,F)=Q$(A,L):NEXTA
12100 IFF<>64THEN12200
12150 GOSUB60050
12200 I1=I1-1
12250 IFJ1<=(R(I1,1)-1)THENJ1=J1+1:GOTO11250
12300 I1=I1-1
12350 IFI1<1THEN12550
12400 J1=R(I1,2)
12450 GOTO12250
12550 I=I-1
12600 D(P(I,J))=D(P(I,J))-1
12650 IFJ<=(P(I,4)-1)THENJ=J+1:GOTO10300
12700 I=I-1
12750 IFI<1THEN13130
12800 J=P(I,5)
12900 D(P(I,J))=D(P(I,J))-1
12950 GOTO12650
13130 IFSN$="SI"THEN13550
13150 IFGR$="NO"AND((ST$="SI"ANDF=64)OR(ST$="NO"ANDF=16))THEN13400
13170 IFGR$="SI"AND((ST$="SI"ANDF=20)OR(ST$="NO"ANDF=5))THEN13400
13200 IFST$="SI"THEN13350
13225 IFGR$="SI"THENGOSUB59100:GOTO13550
13250 GOSUB61050
```

```
13300 GOTO13550
13350 IFGR$="SI"THENGOSUB58050:GOTO13400
13375 GOSUB60050
13400 IFST$="NO"THEN13550
13420 OPEN10,4
13425 IFGR$="SI"THEN13480
13450 PRINT#10,"TOTALE COLONNE=";SPC(2);NC;CHR$(13);CHR$(13);CHR$(13);CHR$(13)
13475 GOTO13500
13480 PRINT#10,"TOTALE GRUPPI =";SPC(2);NC;CHR$(13);CHR$(13);CHR$(13);CHR$(13)
13500 CLOSE10
13550 IFGR$="SI"THEN13580
13560 PRINT:PRINT"     TOTALE":PRINT"     COLONNE=";NC:PRINT
13570 GOTO13600
13580 PRINT:PRINT"     TOTALE":PRINT"     GRUPPI =";NC:PRINT
13600 PRINT"ALTRE ELABORAZIONI":PRINT"CON LO STESSO":INPUT"PROSPETTO";B1$
13650 IF(B1$<>"SI"ANDB1$<>"NO")ORLEN(B1$)=0THEN13600
13700 IFB1$="SI"THEN3000
13750 END
40040 NC=NC+1
40050 IFSN$="NO"ANDST$="NO"THEN40150
40100 IFNC=INT(NC/20)*20THENPRINT"GR. TROVATI=";NC
40150 IFSN$="SI"THEN41250
40200 IFST$="SI"THEN41050
40300 F=NC-INT(NC/5)*5:IFF=0THENF=5
40350 FORA=1TONR:L=P(A,5):BU$(A,F)=M$(A,P(A,L)):NEXTA
40400 IFF<>5THEN40500
40450 GOSUB59100
40500 GOTO41250
41050 F=NC-INT(NC/20)*20:IFF=0THENF=20
41100 FORA=1TONR:L=P(A,5):BU$(A,F)=M$(A,P(A,L)):NEXTA
41150 IFF<>20THEN41250
41200 GOSUB58050
41250 GOTO12550
58050 OPEN10,4
58075 FORA=1TONR
58100 FORG=1TOF:BS$(G)=BU$(A,G):NEXTG
58150 FORG=1TOF:BL%(G)=LEN(BU$(A,G)):NEXTG
58175 IFF>1THENFORG=1TOF-1:PRINT#10,BS$(G);SPC(4-BL%(G)):NEXTG
58200 PRINT#10,BS$(F)
58250 NEXTA
58300 PRINT#10," ":PRINT#10," "
58350 CLOSE10
58400 RETURN
59100 INPUT"** BATTI RETURN **";X$:PRINT:PRINT
59150 FORA=1TONR
59200 S$=""
59250 FORG=1TOF
59300 S$=S$+BU$(A,G)
59350 S$=S$+" ":IFLEN(BU$(A,G))=3THEN59450
59400 FORG2=1TO(3-LEN(BU$(A,G))):S$=S$+" ":NEXTG2
59450 NEXTG
59500 PRINTS$
59550 NEXTA
59600 PRINT:PRINT"GR. TROVATI=";NC
59650 RETURN
60050 FORG=1TO80:BS$(G)=" ":NEXTG
60100 OPEN10,4
60150 FORA=1TONR
60200 G1=0
```

```
60250 FORG=1TOF
60300 G1=G1+1
60350 BS$(G1)=BU$(A,G)
60400 IFG=INT(G/8)*8THENG1=G1+2
60450 NEXTG
60500 FORG1=1TO80:PRINT#10,BS$(G1);SPC(0):NEXTG1
60550 NEXTA
60600 PRINT#10," ":PRINT#10," "
60650 CLOSE10
60700 RETURN
61050 FORG=1TO22:BS$(G)=" ":NEXTG
61075 INPUT"** BATTI RETURN **";X$:PRINT:PRINT
61150 FORA=1TONR
61200 G1=0
61250 FORG=1TOF
61300 G1=G1+1
61350 BS$(G1)=BU$(A,G)
61400 IFG=INT(G/8)*8THENG1=G1+2
61450 NEXTG
61475 S$="":FORG1=1TO21:S$=S$+BS$(G1):NEXTG1
61500 PRINTS$
61550 NEXTA
61600 PRINT:PRINT" C. TROVATE=";NC:PRINT:PRINT:PRINT
61700 RETURN
62050 FH=1
62100 FORHH=1TOH:FH=FH*HH:NEXTHH
62150 RETURN
63150 IF(KB=0)OR(NB=KB)THENCB=1:RETURN
63200 H=NB:GOSUB62050:CB=FH
63250 H=KB:GOSUB62050:CB=CB/FH
63300 H=NB-KB:GOSUB62050:CB=CB/FH
63350 RETURN
```

immaginare a quale stress conduca l'ingrato compito. Certo è che la gran parte di tali gruppi o ricevitorie (e si parla di molte migliaia di unità) hanno il più delle volte preferito ricorrere a tipi di giocata molto più dispendiosi e meno validi,pur di potere utilizzare "schede da sistema", le uniche che consentano la schematizzazione di insiemi colonnari più complessi, e sempre comunque di tipo integrale.

Il nuovo prodotto, applicabile ad una qualsiasi stampante, ma espressamente concepito per la Commodore VC1515, risolve questo gravoso problema. Adesso un piccolo Vic 20, con l'alimentatore automatico Totoprint, può comodamente realizzare il sogno dei grandi sistemisti: la stampa diretta delle schedine.

In pratica lo strumento, ispirato ad una complicata meccanica, prevede il trascinamento guidato delle schedine (una per volta, naturalmente) fino alla stampante. Il posizionamento della stessa viene garantito dalle precise guide di metallo, ed una volta che la schedina è pronta per ricevere la stampa, un'apposita cellula fotoelettrica interrompe il lavoro dell'alimentatore e fa entrare in funzione la stampante. Una volta espulsa la scheda, perfettamente compilata, Totoprint si rimette in moto, immettendo velocemente l'altra scheda e così via. Giunti all'ultimo foglio, il congegno percepisce, attraverso uno speciale micro-interruttore, l'assenza delle schedine, ed impedisce alla stampante di procedere, in attesa del nuovo rifornimento.

Certo, chi più godrà degli enormi vantaggi forniti dall'originale prodotto, sarà il giocatore più tempestivo: fra non molto il Vic 20 potrebbe agevolare talmente grandi masse di utenti che si passerà senza meno al "14" e forse più.

Nel frattempo, che vinca il migliore.

Presentazione del programma

Il TOT, il primo programma "testato" di una serie di ideazioni, di crescente complessità e perfezione che, al momento attuale, arrivano fino al TOT 13. Il TOT 14, in fase di ultimazione, altro non è che una traduzione in linguaggio macchina del TOT 13, per una maggiore velocizzazione delle subroutines più ricorrenti, a tutto vantaggio della rapidità di esecuzione.

Benchè cosciente dell'oggettiva facilità di "copiatura" dei programmi già posti in commercio, l'autore non ha autorizzato la pubblicazione dei listati più complessi (il programma TOT 13, ad esempio, prevede sofisticate selezioni, con condizionamenti multipli, eliminazione dei segni consecutivi, stampa su schedine, ecc.). Ha accettato, invece, la pubblicazione del listato del TOT 9, che presenteremo qui di seguito, e che, come tutti gli altri programmi preparati dal dott. Carchidi, è commercializzato unitamente ad un indispensabile opuscoletto che ne consente un utilizzo ad alto livello anche nei meno esperti. ai meno esperti.

Appena dato il RUN il programma chiede "Numero righe? ", ed attende un INPUT numerico che indichi il numero delle partite considerate. Partita per partita, il programma chiede poi la suddivisione del pronostico mediante l'uso della virgola. In altri termini, occorrerà distribuire il pronostico su tre colonne verticali.

La colonna di sinistra (che conterrà i pronostici che precedono le due virgole) rappresenta solitamente il pronostico di base, mentre quella di centro (che conterrà i pronostici compresi fra le due virgole) e quella di destra (la quale conterrà, appunto, i pronostici a destra delle due virgole) rappresenteranno, nell'ordine, la prima e la seconda varjante.

Attraverso tale ripartizione, l'utente potrà poi richiedere il quantitativo voluto di segni base e delle diverse varianti, ottenendone il calcolo colonnare e lo sviluppo, a video o su stampante, a colonne singole oppure (ove sia possibile) in gruppi di sotto-sistemi integrali.

Tale richiesta, sarà effettuata dall'operatore dando in input i tre quantitativi, in risposta alla domanda "Formula derivata" (o correzione di errori). I tre input, che esprimono, appunto, questa quantità, andranno immessi separati dalla virgola, come già fatto per il pronostico.

Ovviamente è possibile realizzare i sistemi a sezioni, disponendo in prima variante (seconda colonna) le alternative della prima sezione, ed in seconda variante (terza colonna) le alternative della sezione seconda.

A chi abbia una certa familiarità con la sistemistica, il programma TOT 9 offre, pur nella sua essenzialità, opportunità di selezioni molto complesse e sofisticate, anche se non consente la realizzazione delle condizioni incrociate previste dai programmi TOT 12 e TOT 13. Questi programmi, infatti, consentono il filtraggio multiplo del pronostico, applicando contemporaneamente, ad esempio, le correzioni di errori, le sezioni, le formule derivate, l'eliminazione dei segni consecutivi, ecc.

A differenza del programma TOT 9, il cui destinatario è l'utente dilettante, i programmi TOT 12 e TOT 13 possono essere sfruttati adeguatamente solo da un'utenza che abbia dimestichezza con il gioco e che quindi conosca i criteri più sofisticati su cui si costruiscono i sistemi.

E' importante sottolineare che questi programmi (il cui uso è molto facilitato dagli appositi manuali di istruzioni) arrivano a realizzare economie colonnari nell'ordine del 99 per cento ed oltre, pur senza pregiudicare, a condizioni rispettate, la certezza assoluta del "13". Inutile dire che simili sistemi, se realizzati a mano, comporterebbero settimane di lavoro ed elevate possibilità di errori.

E passiamo a considerare il listato del "fratello minore" TOT 9, il cui uso, estremamente semplice, non richiede alcuna particolare preparazione in fatto di sistemi. ▪

IMPARA A PROGRAMMARE CON IL VIC

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

★ DISPENSA N. 3 ★

Ecco dove trovi i Personal Computer Commodore

Distributori Commodore

Liguria
Pirisi Informatica
Piazza Cavour, 19 - 16043 Chiavari
Tel. 0185/30.10.31

Piemonte
Aba Elettronica di Caramia
Via Fossati, 5/C - 10141 Torino
Tel. 011/33.20.65

Lombardia
Homic Personal Computers srl
Piazza de Angeli, 3 - 20146 Milano
Tel. 02/49.88.201

Veneto, Friuli-Venezia Giulia, Trentino-Alto Adige
CO.R.EL. Friuli Computers
Via Mercatovecchio, 28 - 33100 Udine
Tel. 0432/29.14.66

Emilia-Romagna, Marche
S.H.R. srl
Via Faentina 175/A
48010 Fornace Zarattini (Ravenna)
Tel. 0544/46.32.00

Toscana
M.C.S. Spa
Via Pier Capponi, 87 - 50132 Firenze
Tel. 055/57.13.80

Umbria - Alto Lazio
Atlas System srl
Via Guglielmo Marconi, 17 - 01100 Viterbo
Tel. 0761/22.46.88

Lazio
Kiber Italia srl
P.le Asia, 21 - 00144 Roma Eur
Tel. 06/59.16.438

Abruzzo
Pragma System srl
Via Tiburtina, 57 - 65100 Pescara
Tel. 085/50.883

Campania
Computer Market
Parco S. Paolo Isolato, 9 - 80100 Napoli
Tel. 081/76.72.222

Puglia
Maselli per l'ufficio
Via L. Zuppetta, 5 - 71100 Foggia
Tel. 0881/76.111

Puglia
Business Automation System
Largo De Gemmis, 46/B-46/C-48-48/A-48/B
70124 Bari - Tel. 080/22.75.75-22.73.44

Calabria
Sirangelo Computers srl
Via Nicola Parisio, 25 - 87100 Cosenza
Tel. 0984/75.741

Sicilia
Edilcomput Progetti dell'Ing. Giuseppe Carbone
Via La Farina, 141 Is. L
98100 Messina
Tel. 090/29.28.269

Sardegna
S.I.I. - Sistemi Integrati Informatica
Via S. Lucifero, 95 - 09100 Cagliari
Tel. 070/66.37.46

Ecco dove trovi VIC-20

Distributori Commodore

Negozi Expert

Negozi G.B.C.

Negozi Singer

SINGER

Temporex Italiana

Salmoiraghi

La Rinascente

/R

Bit Shop Primavera

Alessandria Via Savonarola, 13
Ancona Via De Gasperi, 40
Arezzo Via Filippo Lippi, 13
Barletta Via Vitrani, 58
Bari Via Capruzzi, 192
Bassano del Grappa Via Jacopo Da Ponte, 51
Bergamo Via F. D'Assisi, 5
Biella Via Italia, 50/A
Bologna Via Brugnoli, 1
Cagliari Via Zagabria, 47
Campobasso Via Mons. II Bologna, 10
Cesano Maderno Via Ferrini, 6
Cinisello Balsamo V.le Matteotti, 66
Como Via L. Sacco, 3
Cosenza Via Dei Mille 86
Cuneo Corso Nizza, 16
Favria Canavese C.so Matteotti, 13
Firenze Via G. Milanesi, 28/30
Foggia Via Marchianò, 1
Forlì P.zza M. Degli Ambrogi, 1
Gallarate Via A. Da Brescia, 2
Genova Via D. Fiasella, 51/r
Genova-Sestri Via Chiaravagna, 10/R
Imperia Via Delbecchi, 32
L'Aquila Via Strada 85,2
Lecco Via L. Da Vinci, 7
Livorno Via San Simone, 31
Lucca Via San Concordio, 160
Macerata Via Spalato, 126
Merano Via Santa Maria del Conforto, 22
Messina Via Del Vespro, 71
Milano Via Jacopo Palma, 9
Milano Viale Certosa, 91
Milano Via Petrella, 6
Milano Via G. Cantoni, 7
Milano P.zza Firenze, 4
Milano Via Altaguardia, 2
Milano V.le Corsica, 14
Mirano-Venezia Via Gramsci, 40
Monza Via Azzone Visconti, 39
Morbegno Via Fabani, 31
Napoli Corso Vittorio Emanuele, 54
Napoli Via Luigia Sanfelice, 7/A
Novara Baluardo Q. Sella, 32
Padova Via Fistomba, 8
Palermo Via Libertà, 4
Parma Via Imbriani, 41
Pavia Via C. Battisti, 4/A
Perugia Via R. D'Andreotto, 49/55
Pescara Via Tiburtina, 264/bis
Pescara Via Trieste, 73
Piacenza Via IV Novembre, 60
Pisa Via XXIV Maggio, 101
Pistoia Via Adua, 350
Potenza Via Mazzini, 72
Pozzuoli Via Pergolesi, 13
Rimini Via Bertola, 75
Roma Via Ponzio Cominio, 46
Roma Via Cerreto Da Spoleto, 23
Roma P.zza San Donà Di Piave, 14
Roma V.le Quattro Venti, 152
Roma Largo Belloni, 4
Savona Via G. Scarpa, 13/r
Sondrio Via Nazario Sauro, 28
Teramo Via Martiri Pennesi, 14
Terni Via Beccaria, 20
Torino Via Chivasso, 11
Torino C.so Grosseto, 209
Torino Via Tripoli, 179
Trento Via Sighele, 7/1
Treviglio Viale Buonarroti, 5/a
Trieste Via F. Severo, 138
Udine Via Tavagnacco, 89/91
Varese Via Carrobbio, 13
Verona Via Pontiere, 2
Viareggio Via Volta, 79
Voghera P.zza Carducci, 11

5 salti condizionati e incondizionati

Il controllo dei livelli di priorità delle istruzioni.

La maggior parte dei problemi richiede che il vostro computer sia programmato in modo da ripetere una o più serie di istruzioni secondo le esigenze del momento. Ciò è possibile per mezzo delle istruzioni di salto (jump).

Una di queste è l'istruzione GOTO (vai-a), che determina il passaggio del controllo al numero di linea dell'istruzione indicata. In altre parole, il computer esegue l'istruzione verso la quale è "saltato" e poi tutte le altre che seguono in sequenza finchè non incontra un'altra istruzione di salto. Ad esempio:

```
50 LET I = 1
60 PRINT I
70 LETI = I + 2
80 GOTO 60
```

determina la stampa dei numeri dispari 1. 3. 5 ecc. Tutte le volte che il computer esegue l'istruzione alla linea 80, salta automaticamente alla linea 60 obbedendo all'istruzione che vi si trova. Passa, quindi, all'istruzione 70. Possiamo quindi dire che GOTO è un istruzione di salto incondizionato, dato che viene eseguito indipendentemente dalle condizioni esistenti.

Avrete notato che nel pezzo di programma riportato non esiste nessuna istruzione che fermi il programma durante la fase di esecuzione, che perciò è destinata ad andare avanti all'infinito.

Per poter fermare il computer durante la fase di esecuzione di questo set di istruzioni, sarà necessario inserire un altro tipo di salto legato all'esistenza di una particolare condizione. Questo tipo di salto, detto condizionato, esegue un test per verificare l'esistenza di una certa condizione e, in base al risultato ottenuto, passa o meno il controllo ad un'altra parte del programma.

In basic, il salto condizionato è rappresentato dall'istruzione IF... THEN (se...allora). Per interrompere, ad esempio, il programma che stampa numeri dispari, potete aggiungere al gruppo di istruzioni riportato più sopra, le due che seguono:

```
65 IFI = 21 THEN 90
90 END
```

Eseguite quindi il programma, verificando se effettivamente esso si interrompe dopo la stampa del numero 21. Se sostituite 21 con un numero pari, diciamo 20 o 22, il programma non si interromperà mai dal momento che I non può avere questo valore.

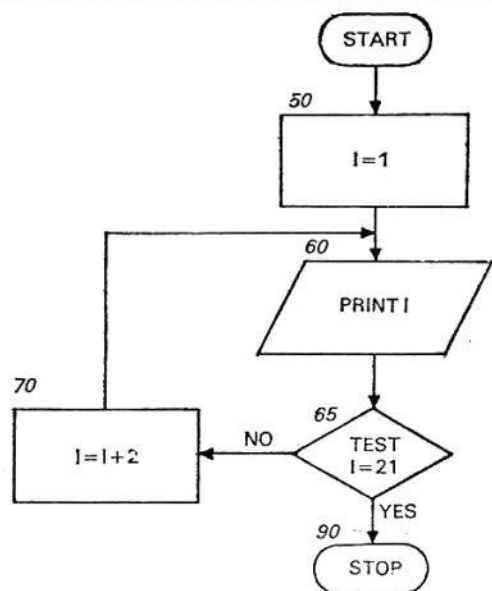
I loop e il loro controllo

Il programmino che avete appena esaminato è costituito da un set di istruzioni, dalla linea 60 alla 80, eseguite in forma ciclica, cioè di loop (loop significa anello, cerchio). Per avere un'idea più chiara di ciò che si intende per loop e per salto fuori dal loop, sarà sufficiente dare un'occhiata allo schema a blocchi riportato nella fig. 5.1. Noterete che l'istruzione GOTO 60 è rappresentata da una freccia che va dal riquadro 70 al riquadro 60.

I modi per uscire da un loop e saltare verso parti diverse di un programma sono numerosi. Osservate il formato dell'istruzione IF ... THEN:
numero di linea **IF** *espressione relazionale* **THEN** *numero di linea diverso.*

Avrete notato che il numero di linea segue la parola THEN deve essere diverso da quello che precede la parola IF, altrimenti è l'istruzione IF stessa a causare il loop.

L'espressione relazionale costituisce la verifica

da eseguire. Se è vera (cioè, se la condizione sussiste), il controllo passa al numero della linea successiva a THEN; se è falsa, passa al numero della linea successiva all'istruzione IF e le istruzioni continuano ad essere eseguite in sequenza finchè non viene raggiunta un'altra istruzione di salto.

L'espressione relazionale confronta due espressioni nel formato seguente:

espressione operatore relazionale *espressione*

Se ricordate, un operatore relazionale l'avete già usato nel capitolo precedente: si trattava dell'operatore =. Per avere comunque una lista completa di tutti gli operatori relazionali disponibili, confrontate la tabella seguente:

operatore relazionale	significato
=	uguale a
>	maggiore di
<	minore di
>=*o*=>	maggiore di o uguale a
<=*o*=<	minore di o uguale a
<>*o*><	non uguale a

Tabella 5.1 Gli operatori relazionali

L'istruzione IF ... THEN è particolarmente utile nell'introduzione dei dati, in cui può essere usata per verificare la correttezza dei risultati finali, individuando la presenza di un eventuale valore finto, cioè di un valore indicante la fine dell'elenco dei dati e che non viene usato per l'esecuzione di calcoli nell'ambito del programma. Il concetto è illustrato nella tabella 5.2 che contiene un programma di somma numerica, in cui i numeri vengono introdotti uno per volta in risposta all'istruzione INPUT della linea 30.

```
10 PRINT"ADD N NUMBERS"
11 PRINT
20 LET T=0
30 INPUT X
40 IF X<=0 THEN 70
50 LET T=T+X
60 GOTO 30
70 PRINT"TOTAL =";T
80 END
```

Tabella 5.2 Una chiusura di programma con un valore falso

Il programma della tabella 5.2 si fermerà quando in X verranno letti o uno zero o un valore negativo. L'istruzione IF ... THEN deve comparire prima dei calcoli che coinvolgono la X, in modo che essi non facciano uso del valore finto.

Un altro modo per interrompere l'esecuzione di un set di istruzioni consiste nello specificare quante volte il loop deve essere eseguito, come indicato nella tabella 5.3.

```
10 INPUT N
20 PRINT"ADD";N;"NUMBERS"
25 PRINT
30 LET I=0
35 LET T=0
40 INPUT X
50 LET T=T+X
60 LET I=I+1
70 IF I<N THEN 40
80 PRINT"TOTAL =";T
90 END
```

tabella 5.3 Programma che esegue la somma di N numeri

Se la linea 30 della tabella 5.3 avesse letto

30 LET I = 1

la linea 70 avrebbe dovuto essere:

70 IF I< = N THEN 40

Questo perchè, quando il loop è stato eseguito N volte, se I è fissato in modo da partire da 1, il

valore di I, dopo che è stata eseguita la linea 60 è maggiore di uno rispetto al numero dei numeri di linea. Ciò significa che il loop termina non appena I = N + 1.

Il confronto fra le stringhe di caratteri

Poichè ogni carattere memorizzato nel computer è rappresentato da un'unica combinazione di cifre binarie, è possibile usare l'istruzione IF ... THEN per effettuare il confronto anche fra stringhe di caràtteri, per esempio, per verificare se P$ contiene il carattere H. L'istruzione

25 IF P$ = "H" THEN 30

significa che la condizione è vera e quindi il salto alla linea 30 può essere effettuato.

Questa capacità del linguaggio è particolarmente utile per confrontare fra loro nomi, indirizzi ed altre informazioni di questo tipo, usate specialmente nelle applicazioni gestionali. Per scoprire quali caratteri hanno un valore superiore o inferiore per i test relazionali "maggiore di" o "minore di", avrete bisogno di potervi riferire ad un elenco di codici usati per rappresentare i caratteri nella memoria del vostro computer.

Le istruzioni FOR ... NEXT

Nei paragrafi precedenti il numero di volte in cui il loop doveva essere eseguito veniva programmato impostando un valore iniziale per il contatore di loop, sottoponendo ad opportuno test il valore finale ed incrementando il valore corrente del contatore di loop se il valore finale non era ancora stato raggiunto. L'istruzione FOR ... NEXT (Per ... quindi), ha lo scopo di semplificare la programmazione di queste tre operazioni.

Nell'esempio della somma di N numeri, nella tabella 5.3, la variabile I (usata come contatore di loop) è stata impostata sul valore iniziale 0, diventato poi 1, dopo la lettura del numero e la sua somma. Terminata questa operazione, è stato eseguito un test (I (N) per stabilire se il programma doveva ritornare all'inizio del loop o doveva fermarsi.

Con l'istruzione FOR ... NXET, il programma può subire delle interessanti modifiche. Ogni istruzione FOR...NEXT è costituita da due linee di codice. All'inizio del loop, per stabilire le condizioni iniziali, l'incremento o passo *(step)* da eseguire alla fine del loop e il valore finale, viene usato l'istruzione FOR in un modo la cui forma può essere sintetizzata nel modo seguente:

numero di linea FOR variabile = espressione 1 TO espressione 2 STEP espressione 3

dove l'espressione 1 stabilisce il valore iniziale del contatore di loop (noto come indice), l'espressione 2, quello finale e l'espressione 3 l'incremento da sommare alla variabile alla fine di ogni esecuzione del set di istruzioni del loop. Se STEP è uguale a 1, sia la parola STEP che l'espressione 3 possono essere tralasciate.

L'istruzione finale del loop si presenta nel formato seguente:

numero di linea NEXT *variabile*

dove la variabile ha lo stesso nome di quello dato nella relativa istruzione FOR.

Il programma riportato nella tabella 5.3 può essere migliorato nel modo indicato nella tabella 5.4. In X viene letto un numero N volte, in base a quanto stabilito dalle istruzioni FOR ... NEXT. Nella linea 35, I viene posta inizialmente a 1, quindi, viene incrementata di 1 nella linea 60 e, se è maggiore di N, il programma va alla linea 80 e stampa il totale, altrimenti ritorna alla linea 40.

```
10 INPUT N
20 PRINT"ADD";N,"NUMBERS"
25 PRINT
30 LET T=0
35 FOR I=1 TO N
40 INPUT X
50 LET T=T+X
60 NEXT I
80 PRINT"TOTAL =",T
90 END
```

Tabella 5.4 Programma di somma alternativo

Inserite l'istruzione

70 PRINT I

in modo che, dopo che il loop è stato eseguito per il numero di volte stabilito, possiate vedere il valore di I.

I può essere usata dentro il loop, facendo però

attenzione a non modificarla (assegnandole cioè un nuovo valore) nel loop, dato che l'operazione produce una variazione delle condizioni stabilite dall'istruzione FOR ... NEXT. Il problema sintetizzato nello schema a blocchi della figura 5.1, può essere codificato come segue:

```
50 FOR I = 1 TO 21 STEP 2
60 PRINT I
70 NEXT I
80 END
```

Il valore dell'incremento dato nell'espressione successiva a STEP può essere negativo (nel qual caso decrementa il contatore) o frazionario. La tabella 5.5 riporta un programma, vale a dire:

```
10 PRINT"START,END AND STEP"
15 PRINT"VARIABLES"
16 PRINT
20 INPUT A,B,C
30 FOR I=A TO B STEP C
40 PRINT I
50 NEXT I
60 END
```

Tabella 5.5 Variabili di avvio, di chiusura e di incremento

in cui è possibile introdurre le variabili di avvio, di chiusura e di incremento, cioè A, B e C. Fate delle prove con un certo numero di combinazioni diverse, compresi i valori negativi e quelli frazionari, e guardate che cosa succede. La tabella 5.6 riporta un programma simile, in cui, però, al posto delle semplici variabili, sono state usate delle espressioni.

```
10 PRINT"START,END AND STEP"
15 PRINT"EXPRESSIONS"
16 PRINT
20 INPUT A,B,C
30 FOR I=A+1 TO B/2 STEP C-3
40 PRINT I
50 NEXT I
60 END
```

Tabella 5.6 Espressioni di avvio, di chiusura e di incremento

L'istruzione ON ... GOTO

L'istruzione ON ... GOTO (nel caso che ... vai-a), si presenta nella forma seguente:
numero di linea ON *espressione* GOTO *due o più numeri di linea separati da virgole.*

La parte intera dell'espressione valutata deve essere un numero positivo non superiore al numero dei numeri di linea che vengono dopo la parte GOTO dell'istruzione.

Il controllo passerà al primo, secondo, terzo.... numero di linea dopo GOTO solo se la parte intera dell'espressione sarà uguale a 1, 2, 3 ...

Per esempio, esistono problemi in cui certi calcoli devono essere eseguiti secondo un dato codice, come in questo caso. Supponete di dover introdurre un certo numero di set di dati, ciascuno dei quali è costituito da un codice (1, 2, 3, 4 o 5) e dai valori di X e di Y.

I calcoli devono essere eseguiti su ciascun set di dati secondo le regole riportate nella tabella 5.7).

Codice	Calcolo
1	R = X + Y
2	R = X - Y
3	R = X * Y
4	R = X/Y
5	R = X Y

Tabella 5.7 Calcoli per codici diversi

A questo punto, potete scrivere un programma che tabuli il codice, i valori di X e di Y e i risultati dei calcoli. Per controllare quale calcolo deve essere eseguito in base al codice relativo, usate l'istruzione ON ... GOTO. Dovrete disegnare uno schema a blocchi del programma, preparare i dati per il test e infine codificare ed eseguire il programma scritto. Ricordate che i dati per il test devono verificare tutti i salti del programma. Per quanto riguarda la sequenza di ingresso, non esiste un ordine preciso, nel senso che, ad esempio, il primo set di dati può avere il codice 3, quello successivo il codice 1, ecc. Confrontate poi il vostro programma con quello riportato nella tabella A3. (Nell'Appendice B, troverete i dati per i test e i valori calcolati più adatti allo scopo).

Un altro modo di impiego della funzione TAB e dei loop FOR

La funzione TAB può essere usata con le varia-

bili o con le espressioni, chiuse fra parentesi e poste dopo la parola TAB, ad esempio: TAB(I), TAB-(P–1). Il programma riportato nella tabella 5.8 genera un rettangolo di dimensioni variabili, costituito da due lati orizzontali tratteggiati (L1) e da due linee verticali formate da frecce verso l'alto (L2).

```
20 PRINT"START COLUMN POSITION"
30 INPUT P
40 PRINT"LENGTH ACROSS & DOWN"
50 INPUT L1,L2
60 LET K=1
70 PRINT TAB(P-1);
80 REM OUTPUT DASHES ACROSS
90 FOR I=1 TO L1
100 PRINT"-";
110 NEXT I
120 PRINT
130 IF K=2 THEN 240
140 REM OUTPUT "↑"S DOWN
150 FOR I=1 TO L2-2
160 PRINT TAB(P-1);"↑";
170 FOR J=1 TO L1-2
180 PRINT" ";
190 NEXT J
200 PRINT"↑"
210 NEXT I
220 LET K=K+1
230 GOTO 70
240 END
```

Tabella 5.8 Programma per la generazione di rettangoli

Qualche osservazione sul programma:

1 Le linee 20 e 40 generano un messaggio per l'utente al quale viene chiesto di introdurre dei dati.

2 L'istruzione PRINT della linea 70 è chiusa da un punto e virgola, il quale fa in modo che la successiva istruzione PRINT eseguita, esca sulla stessa linea.

3 La linea 120 serve a generare la linea tratteggiata. Dopo l'esecuzione della linea 220, che stabilisce che K è uguale a 2, vengono ripetute fino al completamento del rettangolo le linee dal numero 90 al numero 120, dopodichè l'esecuzione del programma ha termine.

4 Le linee dal numero 150 al numero 210 contengono un loop FOR che, a sua volta, ne contiene un altro (linee 170-190).

Questo tipo di loop FOR è detto "annidato" e ne vedremo in dettaglio le caratteristiche nel capitolo 8. Ad ogni passaggio attraverso il loop FOR esterno, quello interno viene eseguito L1-2 volte, in modo che venga generata una freccia seguita da un certo numero di spazi e da un'altra freccia. Non appena viene raggiunta la linea 190, viene eseguito un altro passaggio attraverso il loop esterno finchè non sono state generate L2-2 linee costituite da una freccia, alcuni spazi e una seconda freccia.

In questo caso, l'uso dei loop FOR annidati potrebbe essere evitato sostituendo il contenuto delle linee da 160 a 220 con l'istruzione seguente:

160 PRINT TAB(P-1);" ";TAB(L1-1 +P);" "

Figura 5.2 Rettangolo generato dal programma

La figura 5.2 riporta un rettangolo generato dal programma precedente ed eseguito con questi dati:

i quali stanno a significare che sono stati generati 20 trattini per le due linee orizzontali e 8 frecce per le due verticali.

Per acquisire una certa pratica, vi consigliamo di introdurre il programma riportato nella tabella 5.8 nel vostro computer e di eseguirlo introducendo dati diversi. Usando poi i caratteri grafici per i quattro angoli e i quattro lati, potete scrivere un altro programma che generi un rettangolo di larghezza e altezza variabili al centro dello schermo. Controllate poi nella tabella A4 il risultato.

Esercizi

Gli esercizi riportati fanno riferimento al programma dei nomi e degli indirizzi fornito nella tabella 2.1 a pag. 15.

Esercizio 1. Opzioni di stampa

Modificate il programma della tabella 2.1 in modo da poter effettuare la selezione di una qualsiasi combinazione delle tre opzioni di stampa seguenti:

Codice	*Opzione*
H	intestazione lettera
N	etichetta registrazione
L	etichetta busta
F	arresto esecuzione programma

Controllate poi il risultato nella tabella A5.

Esercizio 2. Intestazioni di lettera

Modificate il programma nella tabella 2.1 in modo da poter introdurre in fase di esecuzione la posizione di stampa dell'intestazione della lettera e il numero di fogli intestati. Controllate poi il risultato nella tabella A6.

Esercizio 3. Etichette di registrazione

Modificate il programma nella tabella 2.1 in modo che il nome di N$ venga generato in posizione centrale all'interno di un riquadro di asterischi. Fate in modo che il numero di etichette richiesto, la lunghezza del nome, e il numero di etichette per pagina da generare possa essere introdotto al momento dell'esecuzione. Controllate poi il risultato nella tabella A7.

Esercizio 4. Etichette per busta

Modificate il programma nella tabella 2.1 in modo da far sì che al momento dell'esecuzione possano essere introdotti i dati relativi al numero di etichette necessario e al numero di etichette per pagina. Controllate poi il risultato nella tabella A8.

Nota: Per ottenere la stampa del numero necessario di etichette per intestazione, per registrazione e per busta, dovrete inserire dei loop nel programma degli esercizi 2, 3 e 4.

Dichiarazione dei redditi

Primavera, tempo di tasse. Questo programma della Bits and Bytes consente la preparazione del modello 104 per la dichiarazione dei redditi.

Il programma, presentato nella versione nastro, gira su tutti i modelli CBM ed è disponibile su supporto magnetico (nastro o dischetto) presso la redazione.

```
5 PRINT"□";:DIMA$(21),K(2,2,21):GOSUB280
6 PRINTTAB(4)AV$:PRINTTAB(5)AP$:PRINTTAB(8)AQ$:PRINTAZ$;"-";
7 PRINTAU$;:F=20:GOSUB200:R$=IN$
10 PRINT:PRINT:PRINT"DICHIARAZ. INDIVIDUALE O CONGIUNTA (I-C)";
11 F=1:GOSUB200:IFIN$="I"THENM1=1:GOTO15
12 IFIN$="C"THENM1=2:GOTO15
13 GOSUB260:GOTO11
15 FORM=1TOM1:PRINT"□";AA$:PRINTAZ$
16 PRINT"N. TIPO DI REDDITO     REDDITI DEDUZIONI":PRINT
20 FORN=0TO10:IFN=4THEN25
21 PRINTRIGHT$(STR$(N+70),2);" ";A$(N)
25 NEXTN:PRINT"▣";SPC(28);B$(M);
26 FORN=0TO10:F=-9:IFN=4THEN45
27 IFN<4THENQ=N+5:GOTO30
28 Q=N+4
30 GOSUB190:IFK(M,1,N)<0THENGOSUB258:GOTO30
35 IFN=1ORN=2ORN=5ORN=6ORN=7THEN40
36 GOTO45
40 GOSUB195:IFK(M,2,N)<0ORK(M,2,N)>K(M,1,N)THENGOSUB258:GOTO40
45 NEXTN
50 GOSUB325
55 IFIN$="N"THEN95
56 IFIN$<>"S"THEN50
60 GOSUB330
65 IFE<70ORE=74ORE>80THEN50
70 N=E-70:F=-9:IFN<4THENQ=N+5:GOTO75
71 Q=N+4
75 GOSUB190:IFK(M,1,N)<0THENGOSUB258:GOTO75
80 IFN=1ORN=2ORN=5ORN=6ORN=7THEN85
81 GOTO90
85 GOSUB195:IFK(M,2,N)<0ORK(M,2,N)>K(M,1,N)THENGOSUB258:GOTO85
90 GOTO50
95 NEXTM:GOSUB320
96 FORM=1TOM1:FORM2=1TO2:FORN=0TO10:K(M,M2,11)=K(M,M2,11)+K(M,M2,N):NEXTN,M2,M
100 FORM=1TOM1:IFK(M,2,11)<=12000000THEN104
101 PRINT"□■■■■■■■■■■■■* TOTALE DEDUZIONI ";B1$(M)
102 PRINT"  SUPERIORI A LIRE 12000000 *":PRINT:PRINT"* RIESAMINARE I CALCOLI"
103 PRINT"  FATTI NEI SINGOLI QUADRI *":FORT=1TO3000:NEXTT:GOTO180
104 NEXTM
105 FORM=1TOM1
110 PRINT"□";AB$:PRINTAZ$
115 PRINT"N. OPERAZ.     DATA    C.BANCA      IMPORTO":PRINT
```

```
120 FORN=12TO13:PRINTRIGHT$(STR$(N+70),2);" ";A$(N):NEXTN
125 PRINT"█";SPC(28);B$(M);:FORN=12TO13:Q=N-7:F=8
126 PRINTLEFT$(AR$,Q);SPC(12);:GOSUB200:C1$(M,N-11)=IN$
127 F=8:PRINTLEFT$(AR$,Q);SPC(21);:GOSUB200:C2$(M,N-11)=IN$
128 F=-9:PRINTLEFT$(AR$,Q);SPC(30);:GOSUB200:GOSUB315:K(M,2,N)=E
129 PRINTLEFT$(AR$,Q);SPC(30);AS$;SPC(10-LEN(STR$(K(M,2,N))));K(M,2,N);:NEXTN
130 GOSUB325
135 IFIN$="S"THEN110
136 IFIN$<>"N"THEN130
140 NEXTM
145 GOSUB320
150 FORM=1TOM1:K(M,2,14)=K(M,1,11):K(M,2,15)=K(M,2,11)
151 K(M,2,16)=K(M,2,14)-K(M,2,15):E=K(M,2,16)*15/100:GOSUB315:K(M,2,17)=E
152 K(M,2,18)=K(M,2,12)+K(M,2,13):K(M,2,19)=K(M,2,17)-K(M,2,18)
155 IFK(M,2,19)<0THENK(M,2,20)=ABS(K(M,2,19)):K(M,2,19)=0:GOTO160
156 K(M,2,20)=0
160 NEXTM
165 PRINT"██QUANTE COPIE DA STAMPARE ";
166 F=-2:GOSUB200:A=E:IFA<0THENGOSUB260:GOTO166
167 IFA=0THEN180
168 PRINT"████TIPO DI STAMPANTE (CBM-HON) ";
169 F=3:GOSUB200:P$=IN$
170 IFP$="CBM"THENP=4:GOTO173
171 IFP$="HON"THENP=5:GOTO173
172 GOSUB260:GOTO169
173 OPEN2,P
175 PRINT"██████PER INIZIARE LA STAMPA,PREMERE █RETURN█ ";:F=0:GOSUB200
176 GOSUB400:FORNC=1TOA:GOSUB335:NEXTNC:CLOSE2
180 PRINT"██ALTRI DATI DA ELABORARE (S-N) ";
181 F=1:GOSUB200:IFIN$="S"THENCLR:GOTO5
182 IFIN$<>"N"THENGOSUB260:GOTO181
185 PRINT"█";:END
190 PRINTLEFT$(AR$,Q);SPC(20);:GOSUB200:GOSUB315:K(M,1,N)=E
191 PRINTLEFT$(AR$,Q);SPC(20);AS$;SPC(10-LEN(STR$(E)));K(M,1,N);:RETURN
195 PRINTLEFT$(AR$,Q);SPC(30);:GOSUB200:GOSUB315:K(M,2,N)=E
196 PRINTLEFT$(AR$,Q);SPC(30);AS$;SPC(10-LEN(STR$(E)));K(M,2,N);:RETURN
200 IN$="":W=0:W1$="":F1=ABS(F)
201 FORW1=1TOF1:PRINT" ";:NEXTW1:FORW1=1TOF1:PRINT"█";:NEXTW1
205 PRINT"_";
210 GETW$:IFLEN(IN$)=F1ANDW$<>CHR$(20)ANDW$<>CHR$(13)ORW$=""THEN210
215 IFW$=CHR$(20)ANDW=0THENPRINT"█";:GOTO200
220 IFW$=CHR$(20)THEN235
225 IFW$=CHR$(13)THEN240
230 PRINT"█";W$;:IN$=IN$+W$:W=W+1:GOTO205
235 PRINT"██  ██";:IN$=LEFT$(IN$,LEN(IN$)-1):W=W-1:GOTO205
240 PRINT"█ ";:E=INT(VAL(IN$)):IFF>0THEN255
245 IFIN$=""THEN255
246 FORW1=1TOLEN(IN$):W1$=MID$(IN$,W1,1)
250 IFASC(W1$)<48ORASC(W1$)>57THENFORW1=1TOLEN(IN$)+1:PRINT"█";:NEXTW1:GOTO200
251 NEXTW1
255 RETURN
258 PRINT"█";:GOSUB260:RETURN
260 IFIN$=""THENPRINT"█ █";:GOTO270
265 FORW=1TOLEN(IN$):PRINT"█  ██";:NEXTW:PRINT"█ █";
270 RETURN
280 A$(0)="D.TERR.(A A-BIS)":A$(1)="AGRARI (A A-BIS)"
281 A$(2)="ALLEVAMENTO (A1)":A$(3)="FABBRIC(B B-BIS)"
285 A$(5)="IMPRESA (F)":A$(6)="IMP.MIN. (G G1)"
286 A$(7)="PARTECIPAZ. (H)":A$(8)="CAPITALE (I)"
287 A$(9)="DIVERSI (L)":A$(10)="TASS.IRP.SEP.(M)"
288 A$(11)="TOTALE REDDITI E DEDUZIONI":A$(12)="VERSAM."
```

```
290 A$(13)="V.INT.":A$(14)="TOTALE REDDITI"
291 A$(15)="TOTALE DEDUZIONI":A$(16)="REDDITI IMPONIBILI"
292 A$(17)="IMPOSTA DOVUTA":A$(18)="ACCONTO VERSATO"
293 A$(19)="IMPOSTA DA VERSARE A SALDO"
294 A$(20)="IMPOSTA DI CUI SI CHIEDE IL RIMBORSO"
295 AA$="RIEPILOGO ILOR":AB$="ACCONTO ILOR":AC$="CALCOLO ILOR A SALDO"
296 FORN=1TO39:AZ$=AZ$+"-":NEXTN:FORN=1TO40:AI$=AI$+" ":NEXTN
297 AR$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
298 AS$="                ▒▒▒▒▒▒▒▒▒▒▒▒"
299 AL$="                               ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
300 FORN=1TO79:AJ$=AJ$+"*":NEXTN:FORN=1TO79:AY$=AY$+"-":NEXTN
302 AU$="RIFERIMENTO DELLA PRESENTE DICHIARAZIONE"
303 AV$="*** MODELLO 740 - QUADRO O ***"
304 AP$="COPYRIGHT 1981  BITS & BYTES":AQ$="MILANO - TEL.02/573344"
305 B$(1)="DICHIARANTE":B$(2)="      CONIUGE"
306 B1$(1)=B$(1):B1$(2)="CONIUGE"
310 RETURN
315 E1=VAL(RIGHT$(STR$(E),3)):IFE1<501THENE=E-E1*SGN(E):GOTO317
316 E=E-(E1-1000)*SGN(E)
317 RETURN
320 PRINT"▒▒▒▒▒▒▒▒▒▒▒";SPC(5);"* ELABORAZIONE DATI IN CORSO *":RETURN
325 PRINTLEFT$(AR$,20);AL$;"CORREZIONI (S-N) ";:F=1:GOSUB200:RETURN
330 PRINTLEFT$(AR$,20);AL$;"QUALE RIGO ";:F=-2:GOSUB200:RETURN
335 S=0:PRINT#2,AV$:PRINT#2," ":PRINT#2,AU$;" :  ";R$
336 PRINT#2,"DICHIARAZIONE :  ";:IFM1=1THENPRINT#2,"INDIVIDUALE":GOTO340
337 PRINT#2,"CONGIUNTA"
340 PRINT#2,AJ$:PRINT#2," ":PRINT#2,AA$;SPC(28);B1$(1);SPC(14);B1$(2)
341 PRINT#2,"NO.";SPC(10);"TIPO DI REDDITO";SPC(11);"REDDITI  DEDUZIONI    ";
342 PRINT#2,"REDDITI  DEDUZIONI":PRINT#2,AY$
345 FORN=0TO11:IFN=4THEN347
346 GOSUB380:GOSUB385
347 NEXTN
350 PRINT#2,AJ$:PRINT#2," "
351 PRINT#2,AB$;SPC(11);A$(12);B1$(1);SPC(11);A$(12);B1$(2)
352 PRINT#2,"NO.  OPERAZIONE";SPC(6);"DATA   COD. BANCA   IMPORTO    ";
353 PRINT#2,"DATA   COD. BANCA   IMPORTO":PRINT#2,AY$
355 FORN=12TO13
356 GOSUB380:PRINT#2,SPC(14-LEN(A$(N)));
360 PRINT#2,C1$(1,N-11);SPC(10-LEN(C1$(1,N-11)));C2$(1,N-11);
361 PRINT#2,SPC(19-LEN(C2$(1,N-11))-LEN(STR$(K(1,2,N))));K(1,2,N);
362 PRINT#2,SPC(1);C1$(2,N-11);SPC(10-LEN(C1$(2,N-11)));C2$(2,N-11);
363 PRINT#2,SPC(19-LEN(C2$(2,N-11))-LEN(STR$(K(2,2,N))));K(2,2,N):NEXTN
364 FORN=12TO13:FORM=1TO2:IFK(M,2,N)>0THENS=S+1
365 NEXTM,N:PRINT#2,SPC(5);"SI ALLEGANO N.";S;"ATTESTATI DI ";A$(12)
366 PRINT#2,AJ$:PRINT#2," "
367 PRINT#2,AC$;SPC(26);B1$(1);SPC(4);B1$(2):PRINT#2,AY$
370 FORN=14TO20:GOSUB380:PRINT#2,SPC(52-LEN(A$(N))-LEN(STR$(K(1,2,N))));
371 PRINT#2,K(1,2,N);SPC(10-LEN(STR$(K(2,2,N))));K(2,2,N):NEXTN
375 PRINT#2,AJ$:PRINT#2,SPC(9);"ELABORATO CON SOFTWARE BITS & BYTES -";
376 PRINT#2," MILANO - TEL.02/573344":FORN=1TO25:PRINT#2," ":NEXTN:RETURN
380 PRINT#2,RIGHT$(STR$(N+70),2);" - ";A$(N);:RETURN
385 PRINT#2,SPC(41-LEN(A$(N))-LEN(STR$(K(1,1,N))));K(1,1,N);
386 IFN=1ORN=2ORN=5ORN=6ORN=7ORN=11THENPRINT#2,SPC(10-LEN(STR$(K(1,2,N))));
387 IFN=1ORN=2ORN=5ORN=6ORN=7ORN=11THENPRINT#2,K(1,2,N);:GOTO390
388 PRINT#2,SPC(11);
390 PRINT#2,SPC(10-LEN(STR$(K(2,1,N))));K(2,1,N);
391 IFN=1ORN=2ORN=5ORN=6ORN=7ORN=11THENPRINT#2,SPC(10-LEN(STR$(K(2,2,N))));
392 IFN=1ORN=2ORN=5ORN=6ORN=7ORN=11THENPRINT#2,K(2,2,N):GOTO395
393 PRINT#2," "
395 RETURN
400 A$(0)="DOMINICALI TERRENI (A - A BIS)":A$(1)="AGRARI (A - A BIS)"
405 A$(2)="DI ALLEVAMENTO (A1)":A$(3)="DEI FABBRICATI (B - B BIS)"
410 A$(5)="DI IMPRESA (F)":A$(6)="DI IMPRESA MINORE (G - G1)"
415 A$(7)="DI PARTECIPAZIONE (H)":A$(8)="DI CAPITALE (I)"
420 A$(10)="A TASSAZIONE IRPEF SEPARATA (M)":A$(12)="VERSAMENTO "
425 A$(13)="VERS. INT.":RETURN
```

Un vocabolario per i dialetti del basic

Il PET è un computer universale. A volte, però, può essere utile convertire per questa macchina alcuni programmi originariamente scritti nei dialetti basic di altri personal. Ecco il primo d'una serie di articoli scritti per illustrarvi come effettuare tale conversione.

PUO' capitare a volte di veder "girare", su personal computer diversi dai modelli Commodore, alcuni programmi che potrebbero esserci utili.

Dovrebbe esser noto che non è possibile registrare tali programmi su di un supporto magnetico e trasferirli in seguito sul PET o sul VIC. Si sa infatti che le procedure di registrazione su nastro o mini-floppy cambiano in modo radicale a seconda della marca del computer pur se il linguaggio utilizzato, quasi sempre il Basic, è lo stesso.

Una cassetta contenente un programma scritto da un Apple

non viene accettato, tanto per fare un esempio, da un sistema CBM o da un Atari o da un Tandy. Ecco quindi perchè, tra l'altro, prima di acquistare un computer, è bene documentarsi sulla reale popolarità e conseguente diffusione di quella marca, proprio per evitare di ritrovarsi in casa una macchina che presenta forse il vantaggio di un basso costo iniziale, ma grosse difficoltà di approvvigionamento di programmi che evitano la fatica di digitarli.

I felici possessori di modelli Commodore sanno bene che è piuttosto semplice incontrarsi con numerosi appassionati disposti a scambiare, vendere e (perchè no?) regalare programmi a chi ne ha bisogno.

Il personal computer più diffuso, dopo il Commodore ovviamente, è l'Apple che, pur se utilizza il Basic Microsoft, presenta alcune istruzioni in più ed alcune differenze di interpretazione che qui di seguito riporto per consentire di far girare sui PET-CBM i programmi Basic dell'Apple.

• *1/ Statement con nessuna differenza di significato o uso nelle due macchine.*
ABS(X) ASC(Stringa) ATN AND CHR$(X) CONT COS DATA DEFFN(X) DIM END EXP FOR ...TO...STEP FRE(X) GET GOSUB GOTO IF...GOTO IF... THEN INPUT INT(X) LEFT$ LEN LET LIST LOAD LOG MID$ NEW NEXT ON...GOSUB ON...GOTO OR POS(X) PRINT READ REM RESTORE RETURN RIGHT$ RND RUN SAVE SGN SIN SPC SQR STEP STOP STR$ TAB TAN VAL(Stringa) WAIT.

Apple	PET
CLEAR	CLR
Ctrl C	RUN/STOP (Tasto)
HOME	PRINT " " (Tasto CLR/HOME)
IN#	INPUT#
↔	(Shift+) CRSR ↔
INVERSE	PRINT "R" (Tasto REVERSE)
NORMAL	PRINT "–" (Tasto OFF REVERSE)
PR#	PRINT#
RECALL	GET# INPUT#
↔	CRSR ↔
STORE	PRINT#

• *2/ Statement che, pur possedendo la stessa sintassi, provocano effetti profondamente diversi nelle due macchine:*
PEEK(X) POKE X,Y USR(X)

La prima di queste tre restituisce il valore contenuto in una particolare locazione di memoria. Molto spesso, nei programmi Basic di una certa complessità, l'istruzione PEEK si riferisce a locazioni utilizzate dal Sistema Operativo (O.S.) che risulta completamente diverso da macchina a macchina e che indica, per esempio, se e quale tasto è stato premuto, se una particolare cella del video è vuota o contiene un carattere, eccetera.

Per lo stesso motivo l'effetto di un POKE, che scrive un valore in una locazione di memoria e quello di un USR(X), che fa partire un programma scritto in linguaggio macchina può risultare disastroso.

E' opportuno allora che solo i lettori più esperti si cimentino nella trasposizione di programmi da Apple a PET che contengano anche una sola delle istruzioni descritte e solo a patto che posseggano la mappa della memoria dell'Apple per individuare l'esatto significato delle tre istruzioni prima descritte.

• *3/ Comandi che, con differente sintassi, provocano lo stesso effetto.*

• *4/ Comandi non disponibili sul Pet e che non possono essere sostituiti da semplici routine emulative scritte a parte:*
COLOR DRAW GR HCOLOR HGR HGR2 ON ERR GOTO POP RESUME SPEED

• *5/ Comandi non disponibili sul PET ma che possono essere sostituiti da semplici routine* che saranno tra breve presentate su Computer Club:
CALL DEL HLIN HPLOT HTAB REPEAT VLIN VTAB FLASH PDL SCRN(X,Y).

• *6/ Comandi disponibili sul PET fornito di scheda grafica:*
DRAW PLOT SCALE XDRAW, eccetera.

• *7/ Comandi disponibili sul PET fornito di ROM speciali o di programmi oem* commercializzate da ditte specializzate:
NOTRACE TRACE

• *8/ Comandi non necessari sul PET in quanto automaticamente eseguiti dal Sistema Operativo:*
HIMEM LOMEM TEXT.

Conversione in base

QUANTE volte vi siete trovati ad eseguire dei calcoli di conversione da una base ad un'altra? Caso tipico è la conversione da esadecimale a decimale o viceversa.

Certamente moltissime volte

In commercio esiste una calcolatrice della Texas che, oltre a fornire la possibilità di calcolo " + – x :", può convertire qualsiasi numero espresso in una qualsiasi base prevista in un altro numero in altra base. Le basi previste sono le solite: decimale, esadecimale, ottale e binaria.

Con questo programma ho voluto evitare sia la tediosa ricerca di questa rara macchinetta ed al tempo stesso risparmiare denaro. Infatti questa T.I. costicchia.

Il programma in oggetto prevede un'interessantissima routine che traduce un numero di qualsiasi base a qualsiasi altra base. Però, in riga 310, limito il numero delle conversioni a quelle direttamente richieste dallo scopo prefisso. Nulla ci vieta, però, modificare il valore della variabile BI% (Base Input) che ho fissato a 10 ed i valori della variabile BO% (Base Output) a cui ho fornito di volta in volta i valori necessari (16,8,2 esadecimale, ottale, binario).

Grazie all'opzione di stampa è possibile ottenere una hardcopy della lista delle conversioni in maniera tale da poter archiviare a mo' di tabellina gli utili fogli di carta pronti in ogni momento alle future consultazioni.

L'esecuzione del programma è a ciclo continuo ed infinito, sarà quindi vostra la scelta del quando interrompere le elaborazioni.

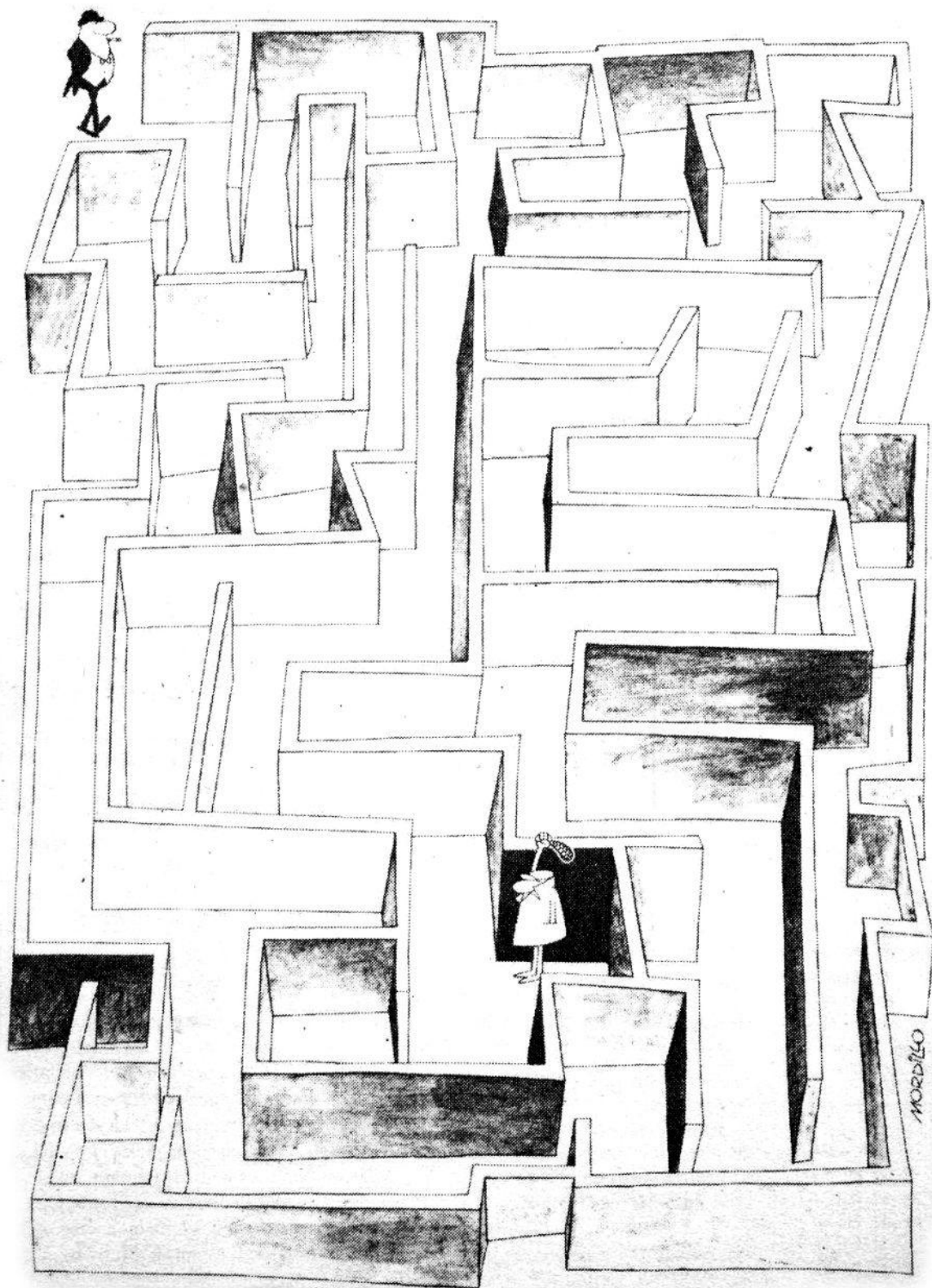
La routine di stampa prevede l'utilizzo delle OPEN di formattazione di maschera di riga, fatto questo ben poche volte utilizzato su questo tipo di stampante. Già che c'è, approfittiamone.

Chi invece non fosse possessore di una 3022 o 4022 potrà stampare le singole variabili sfruttando il medesimo principio dell'evidenziazione sullo schermo; osservate a questo scopo le righe 330 e 340.

La routine di conversione in base potrà essere utilizzata in altri programmi, basta sapere che:
NI$ = numero di input espresso in stringa;
BI% = valore numerico della base di input;
BO% = valore numerico della base di output;
GOSUB 360 = richiamo alla subroutine;
NO$ = numero di output espresso nella base di output;

Potrebbe essere interessante modificare il presente programma in un altro che preveda la possibilità di calcolare e trasformare un qualsiasi numero espresso in una qualsiasi base in un altro espresso in una qualsiasi altra base.

Divertente, istruttivo, utile. No?

```
110 ************************************************************
120 *                                                          *
130 *   CONVERSIONE E PRINT DA NUMERO DECIMALE A NUMERO IN BASE  *
140 *              ESADECIMALE     OTTALE       BINARIO         *
150 *           SU VIDEO E SU STAMPANTE TIPO 3022 O 4022        *
160 *                                                          *
170 ********************************************************KH**
180 PRINT"□□□□□□□□□□□           VUOI LA STAMPA ? ";
190 GETP$:IFP$="S"THENGOTO220
200 IFP$="N"THENP$="":GOTO270
210 GOTO190
220 OPEN2,4,2:OPEN1,4,1:OPEN3,4,3
230 OPEN4,4:PRINT#4,"□":CLOSE4
240 F$="□| ZZZZZ  □=   AAAA  □|   AAAA  □| 9999999999999999 □|"
250 PRINT#2,F$
260 PRINT#3,27:CLOSE3
270 PRINT"□"
280 BI%=10
290 NX=NX+1:NI$=STR$(NX)
300 NI$=RIGHT$(NI$,LEN(NI$)-1)
310 BO%=16:GOSUB360:N1$=NO$:BO%=8:GOSUB360:N2$=NO$:BO%=2:GOSUB360
320 IFP$<>""THENPRINT#1,VAL(NI$);N1$;CHR$(29);N2$;CHR$(29);VAL(NO$)
330 PRINTRIGHT$("     "+NI$,5),RIGHT$("    "+N1$,4),;
340 PRINTRIGHT$("    "+N2$,4),RIGHT$("          "+NO$,10)
350 GOTO280
360 NO$="":REM BASE CONV SOUBROUTINE
370 REM ** CONVERSIONE DECIMALE **
380 L%=LEN(NI$)
390 DEC=0
400 PWR%=0
410 FORJ=L%TO1STEP-1
420 K%=ASC(MID$(NI$,J,1))
430 IFK%>64THENK%=K%-7
440 K%=K%-48
450 IFK%<BI%ANDK%>-1THEN500
460 PRINT"INPUT INVALIDO PER QUESTA BASE";
470 PRINTBI%
480 NO$="????"
490 RETURN
500 DEC=DEC+K%*BI%↑PWR%
510 PWR%=PWR%+1
520 NEXT J
530 REM ** CONVERSIONE IN BASE **
540 H$="0123456789ABCDEF"
550 NC$=""
560 PWR%=LOG(DEC)/LOG(BO%)
570 FORJ=PWR%TO0STEP-1
580 XX=INT(BO%↑J)
590 CH%=DEC/XX
600 NO$=NO$+MID$(H$,CH%+1,1)
610 DEC=INT(DEC-CH%*XX)
620 NEXT J
630 RETURN
640 END
```

MORDILLO

Labirinto

Questo programma gira sul Vic 20 senza bisogno di espansioni di memoria. Il gioco consiste nel cercare di percorrere tutto il labirinto in un tempo massimo di un minuto e trenta secondi.

Lungo il percorso si incontrano dei puntini e dei cerchi che valgono rispettivamente 10 e 100 punti. Il punteggio massimo raggiungibile è 4980.

All'inizio del gioco appare al centro dello schermo una pallina nera; i tasti per muoverla nelle quattro direzioni sono: "P", "L", ";", ".".

Ed ecco ora alcune spiegazioni sul listato per eventuali modifiche:

- la linea 6 pone gli indirizzi di inizio della mappa dei caratteri e dei colori;
- la 710 pone il tempo massimo a un minuto e trenta secondi;
- il sottoprogramma 800 crea una piccola attesa. Modificandolo o addirittura eliminandolo si può variare la velocità della pallina.

Giampaolo Cervone
Via Vincenzo Riolo, 10
90139 Palermo - Tel. 580061


```
2 PM=0:POKE36879,29
3 P=0
6 CA=7680:CO=38400
9 GOSUB1700:GOSUB1000:GOSUB1500
10 I=CA+252:J=CO+252
20 POKEI,81:POKEJ,0
25 TI$="000000"
100 IFPEEK(I-22)=102GOTO120
105 IFPEEK(I-22)=46THENP=P+10:F=0:GOSUB3000
106 IFPEEK(I-22)=87THENP=P+100:F=1:GOSUB3000
110 POKEI,32:I=I-22:J=J-22:POKEI,81:POKEJ,6
112 GOSUB800
120 GETA$:IFA$=";"GOTO200
122 IFA$="L"GOTO300
126 IFA$="."GOTO400
128 GOSUB700
130 GOTO100
200 IFPEEK(I+1)=102GOTO220
205 IFPEEK(I+1)=46THENP=P+10:F=0:GOSUB3000
206 IFPEEK(I+1)=87THENP=P+100:F=1:GOSUB3000
210 POKEI,32:I=I+1:J=J+1:POKEI,81:POKEJ,6
212 GOSUB800
220 GETA$:IFA$="P"GOTO100
222 IFA$="L"GOTO300
226 IFA$="."GOTO400
228 GOSUB700
230 GOTO200
300 IFPEEK(I-1)=102GOTO320
305 IFPEEK(I-1)=46THENP=P+10:F=0:GOSUB3000
306 IFPEEK(I-1)=87THENP=P+100:F=1:GOSUB3000
310 POKEI,32:I=I-1:J=J-1:POKEI,81:POKEJ,6
312 GOSUB800
320 GETA$:IFA$="P"GOTO100
322 IFA$=";"GOTO200
326 IFA$="."GOTO400
328 GOSUB700
330 GOTO300
400 IFPEEK(I+22)=102GOTO420
```

```
405 IFPEEK(I+22)=46THENP=P+10:F=0:GOSUB3000
406 IFPEEK(I+22)=87THENP=P+100:F=1:GOSUB3000
410 POKEI,32:I=I+22:J=J+22:POKEI,81:POKEJ,6
412 GOSUB800
420 GETA$:IFA$="P"GOTO100
422 IFA$=";"GOTO200
426 IFA$="L"GOTO300
428 GOSUB700
430 GOTO400
700 PRINT"■■";:PRINTTAB(16)MID$(TI$,4,1)":"RIGHT$(TI$,2)
710 IFVAL(TI$)>129GOTO750
720 RETURN
750 POKE36878,15:POKE36876,200:FORT=1TO500:NEXT
755 IFPM<PTHENPM=P
760 POKE36878,0:POKE36876,0:PRINT"■";:PRINTTAB(13)"■HI:"PM
765 PRINT"■ PREMI F1 PER GIOCARE "
770 GET X$:IFX$<>"■"THEN770
775 GOTO3
800 FORT=1TO50:NEXT:RETURN
1000 FORK=22TO505
1010 GOSUB2500:NEXT
1020 RETURN
1500 READK:IFK=1000THEN1526
1520 GOSUB2000
1525 GOTO1500
1526 READK:IFK=1000THENRESTORE:RETURN
1527 GOSUB2200
1529 GOTO1526
1530 DATA46,47,48,49,50,58,59,60,72,74,75,76,78,80,82,83,84,85,86,89
1531 DATA90,91,92,94,96,98,100,102,104,106,108,111,118,120,122,124,126
1532 DATA128,133,134,135,136,138,140,142,144,146,148,152,155,158,160,162
1533 DATA164,166,168,170,172,177,178,179,180,182,184,186,188,190,192,196,199,202
1534 DATA203,204,205,206,208,210,211,212,213,216,218,221,222,223,224,230,236
1535 DATA240,246,247,248,249,250,251,252,253,255,256,257,258,260,262,265,266,274
1536 DATA280,281,282,283,284,289,290,291,292,293,294,296,298,300,302,309,310,311
1537 DATA316,320,322,324,325,326,327,328,331,334,336,337,338,340,342,344,346
1538 DATA350,353,355,362,364,366,368,370,371,375,378,379,380,381,382,383,384,385
1539 DATA386,387,388,389,390,391,392,397,399,406,408,414,416,419,421,422,423,424
1540 DATA425,426,428,430,432,433,434,435,436,438,441,460,463,464,465,466,467
1541 DATA468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,1000
1542 DATA45,52,56,93,64,99,116,117,130,149,150,174,194,214,217,238,254,267,318,3
21
1543 DATA333,335,372,377,415,427,431,446,456,1000
1700 FORH=0TO21:POKECA+H,127:POKECO+H,0:NEXT:RETURN
2000 POKECA+K,46:POKECO+K,0:RETURN
2200 POKECA+K,97:POKECO+K,0:RETURN
2500 POKECA+K,102:POKECO+K,2:RETURN
3000 POKE36878,15:POKE36877-F,240:FORT=1TO40:NEXT:POKE36877-F,0
3010 PRINT"■■ PUNTI:"P
3020 IFP=4980THENPRINT"■■■■■■■■ R A V I S S I M O ":GOTO755
3100 RETURN
```

ABA: la microinformatica, chiavi in mano.

ABA ELETTRONICA non si limita a trattare la più ampia gamma di marche e di modelli per tutte le applicazioni, da quelle hobbistiche alle gestionali. ABA ELETTRONICA mette a vostra disposizione il mondo della microinformatica, dai corsi di istruzione a vari livelli, all'assistenza tecnica più qualificata, alla vendita di periferiche, accessori e pubblicazioni. Vi aiuta a scegliere inoltre. Nella sua sala di dimostrazione è possibile provare e confrontare quanto di meglio offre oggi il mercato. E quando avrete deciso per un microcomputer, ABA ELETTRONICA vi propone di scegliere la forma di acquisto che preferite. Anche in leasing o per corrispondenza.
Infine ABA ELETTRONICA vi fornisce tutti i programmi, standard o su misura, siano essi gestionali, professionali o scientifici che Vi necessitano provvedendo anche all'addestramento dell'operatore sul sistema che avete scelto e su tutta la microinformatica che lo riguarda.
Chiavi in mano.

Quella del Commodore, ad esempio.

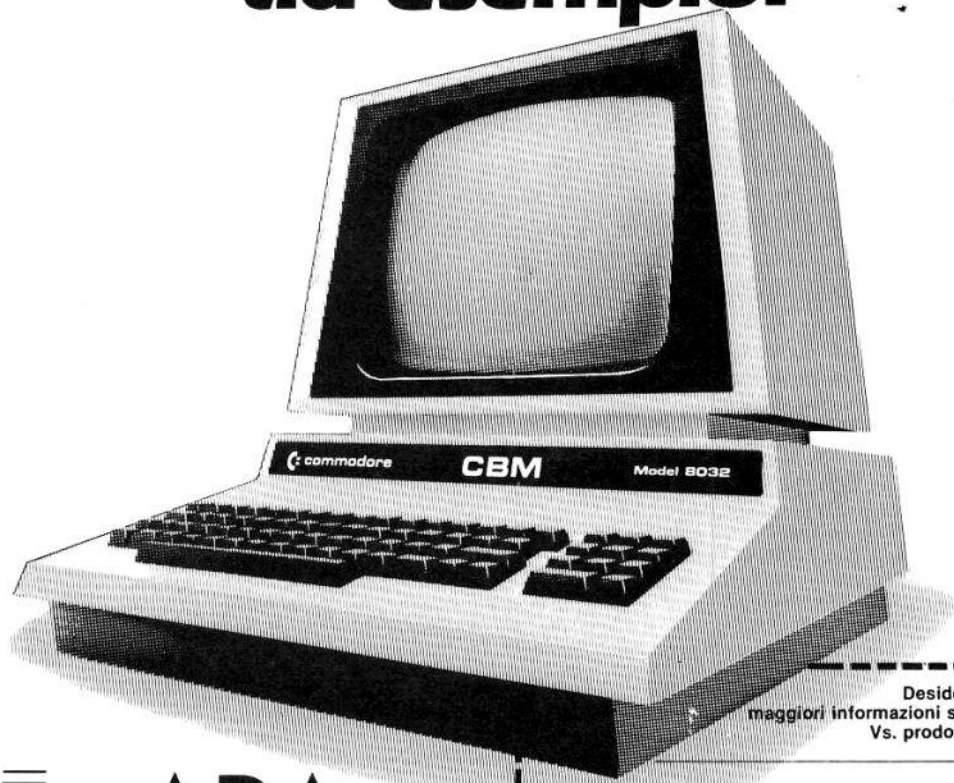
commodore CBM Model 8032

Desidero ricevere maggiori informazioni sui seguenti Vs. prodotti e servizi:

Nome

Cognome

Via

Città Telefono

ABA ELETTRONICA

Il centro più completo a memoria di computer.

Vendita, Programmazione e Assistenza:
ABA ELETTRONICA - 10141 Torino - Via Fossati 5/c
Tel. (011) 33.20.65/38.93.28

VIC 20 un vero sistema con una bella sorpresa anche nel prezzo

Listino prezzi del VIC 20 e di alcuni accessori

CODICE	PRODOTTO	PREZZO LIRE (IVA esclusa)
	VIC 20 UN SISTEMA COMPLETO	
VIC 20	**Home Computer** Memoria base 5K, 255 combinazioni di colori, 3 voci, BASIC residente, si collega direttamente a qualsiasi televisore	423.000
VIC 1530	**Registratore a cassette** Per memorizzare facilmente programmi e dati su normali cassette magnetiche	120.000
VIC 1515	**Unità stampante** Stampa velocemente su carta normale quanto appare sul video: programmi, lettere, dati, grafici	650.000
VIC 1540	**Unità a disco (Floppy Disk)** Veloce unità di memoria di massa ad alta capacità. Può immagazzinare fino a 170.000 caratteri su ogni singolo disco	680.000
	MEMORIA D'UN GRANDE COMPUTER	
VIC 1210	**Cartuccia da 3K di memoria** Espande la memoria RAM del VIC di 3K	66.000
VIC 1110	**Cartuccia da 8K** Espande la memoria RAM del VIC di 8K	98.000
VIC 1111	**Cartuccia da 16K** Espande la memoria RAM del VIC di 16K	172.000
	È FACILE PROGRAMMARE CON IL VIC 20	
VIC 2013	**VIC FORTH** Con questo nuovo linguaggio il VIC 20 si potenzia con una gamma infinita di possibilità (Cartuccia)	95.000
VIC 2501	**Introduzione al BASIC** - Parte prima	24.500
VIC 2502	**Introduzione al BASIC** - Parte seconda Un manuale dettagliato in due parti per apprendere il linguaggio BASIC. Ogni parte è corredata da un volume e da 2 cassette con numerosi esempi	24.500
	LA GIOIA DEI VIDEOGIOCHI	
VIC 1901	**Avenger (I vendicatori)** Gli invasori spaziali stanno assaltando la terra. L'astronave si difende sparando ed evitando di essere colpita	41.000
VIC 1902	**Star Battle (Guerre Stellari)** Una serie di astronavi nemiche attacca in forze. Rapidità e precisione sono necessarie per la vittoria	41.000
VIC 1904	**Super Slot (Slot machine)** Simile alle macchine «mangiasoldi» di Las Vegas, ha effetti musicali e sonori di grande impatto	41.000
VIC 1905	**Jelly Monsters (I mostri di gelatina)** Dei mostri-fantasmi inseguono il giocatore. Bisogna completare il percorso senza essere presi	41.000
VIC 1906	**Alien (Gli extraterrestri)** Quattro extraterrestri contendono il labirinto all'astronauta. Solo la loro eliminazione porta alla salvezza	41.000
VIC 1907	**Super Lander (Atterraggio lunare)** L'astronave deve atterrare su un pericoloso pianeta. Per una manovra perfetta ci vuole un ottimo pilota	41.000
VIC 1908	**Poker** Come al casinò è possibile vincere o perdere grandi fortune. Disegni dettagliati ed effetti sonori danno un aspetto di vero realismo	41.000
VIC 1909	**Road Race (Corsa automobilistica)** Riflessi pronti ed abilità di guida sono indispensabili per non uscire di strada	41.000
VIC 1910	**Rat Race (Corsa del topo)** Il percorso è pieno di gatti e di pericoli. Solo un topolino in gamba può mangiarsi tutte le porzioni di formaggio	41.000
VIC 1912	**Mole Attacks (L'invasione delle talpe)** Ci sono sei buchi nel terreno e le talpe fuoriescono in continuazione. Non è facile impedirlo con un solo giocatore ma non è impossibile	41.000
VIC 1919	**Sargon Chess (Scacchi)** Secondo gli esperti è un'ottimo programma di scacchi provvisto di sei livelli di difficoltà; il VIC è uno sfidante eccezionale	41.000
T-1	**Interfaccia IEEE 488** per VIC 20	175.000

Prezzi in vigore dal 31/3/83 e soggetti a modifica senza priore notifica.

Corde del mio computer

Volete crearvi un archivio elettronico per memorizzare gli accordi di chitarra più usati. Ecco il programma che fa per voi.

COME è noto a tutti coloro i quali suonano o hanno suonato una chitarra ritmica, gli accordi, sugli spartiti, non sempre sono rappresentati sulla classica intavolatura con le relative posizioni delle dita. Diventa così necessario ricorrere a costosi libri di accordi nei quali la ricerca di qualche cosa di utile è spesso lunga e difficoltosa a causa del gran numero di posizioni descritte.

Ecco dunque un programma semplice ma (a parere mio e di quelli che l'hanno usato) molto utile e chiaro soprattutto nella presentazione grafica dell'accordo ottenuta sfruttando i caratteri grafici del Vic.

Il programma non fa che riempire le stringhe dati con la codifica dell'accordo, dopo di che chiede il nome dell'accordo da cercare.

Dovendo far stare ben 132 accordi in soli 3500 byte s'è dovuta sfruttare la notazione anglosassone molto più sintetica, così il Do = C; Re = D; Mi = E; Fa = F; Sol = G; La = A; Si = B.

Per quanto riguarda le tonalità tra quelle fondamentali sono riportati solo i diesis perchè già presenti sulla tastiera. I bemolle possono comunque sempre essere trasformati in diesis come ben sanno coloro che di musica se ne intendono appena un poco.

Se l'accordo è presente in memoria viene disegnata la tastiera con la posizione delle dita e le indicazioni di quali dita usare. In caso contrario viene segnalata la mancanza del nominativo in memoria e si passa ad un nuovo input.

Il codice di memorizzazione è il seguente: per ogni accordo viene data una serie di dodici cifre. Di esse la prima di ogni coppia rappresenta il numero del dito (0 = corda libera, X = corda da non suonare, dall'uno al quattro rispettivamente indice, medio, anulare e mignolo), mentre la seconda indica a che tasto deve apparire il punto nero indicante la posizione del dito.

Per chi avesse molta memoria a disposizione (es. espansione grafica o 16 K) si consiglia di aggiungere anche altri accordi oppure di cambiare quelli dati. Buon Concerto a tutti. ▪

Alessandro Cattani
Via P. Teulliè, 11 - 20136 Milano
Tel. 8397776

```
1 REM*********************
2 REM*
3 REM*  ALESSANDRO CATTANI
4 REM*
5 REM*      132  ACCORDI
6 REM*
7 REM*********************
10 K=131:GOSUB120
20 INPUT"]ACCORDO (IN INGLESE)";A$:PRINT"]▒▒  ACCORDO = ";A$
30 PRINT"▒▒":PRINT" ┌┬┬┬┬┬┬┬┬┐":A1$=" | | | | | | | | |":A2$=" ├┼┼┼┼
┼┼┼┤"
40 FORT=1TO4:PRINTA1$:PRINTA2$:NEXT:PRINTA1$:PRINT" └┴┴┴┴┴┴┴┴┘":F=0
50 IFF>KTHENPRINT"]▒▒▒   NON MEMORIZZATO":FORT=1TO1000:NEXT:GOTO20
60 IFRIGHT$(C$(F),LEN(C$(F))-12)<>A$THENF=F+1:GOTO50
70 PRINT"▒▒▒▒▒▒":B$=C$(F)
80 FORT=1TO12STEP2
81 Z$=MID$(B$,T,1)
82 PRINTTAB(0)Z$
83 IFZ$="X"ORZ$="0"THENPRINT"▒";:NEXT:GOTO96
84 PRINT"]";:PRINTTAB(VAL(MID$(B$,T+1,1))*2)"●"
95 PRINT"▒";:NEXT
96 PRINT"▒▒PER CONTINUARE PREMERE LA BARRA SPAZIATRICE "
100 GETD$:IFD$<>" "THEN100
110 GOTO20
120 DIMC$(K):FORT=0TOK:READC$(T):NEXTT:RETURN
```

```
121 DATA0011002233XXC,XX11004333XXC4,XX11112233XXC5+,XX45352413XXC5-,1324453513X
XCM
122 DATAXX45133513XXC7,1122113344XXC#,XX47362614XXC#4,XX12122334XXC#5+,XX4636251
4XXC#5-
123 DATA1425463614XXC#M,XX46143614XXC#7,22331200XXXXD,33231200XXXXD4
124 DATAXX13132435XXD5+,22331100XXXXD5-,11432200XXXXDM,32112200XXXXD7
125 DATA33442311XXXXD#,44342311XXXXD#4,XX14142536XXD#5+,33442211XXXXD#5-
126 DATA22443311XXXXD#M,43223311XXXXD#7,000011322200E,000042322200E4,00211132XXX
XE5+
127 DATA34452312XXXXE5-,000000322200EM,44233412XXXXE7,111122433311F,111143332311
F4
128 DATA11322243XXXXF5+,35462413XXXXF5-,111111433311FM,45243513XXXXF7,1212234434
12F#
129 DATA121244342412F#4,12332344XXXXF#5+,36472514XXXXF#5-,121212443412F#M
130 DATA46253614XXXXF#7,131324453513G,131345352513G4,13342445XXXXG5+,37482615XXX
XG5-
131 DATA131313453513GM,47263715XXXXG7,141425463614G#,141446362614G#4,14352546XXX
XG#5+
132 DATA24133546XXXXG#5-,141414463614G#M,141425143614G#7,151526473715A,151547372
715A4
133 DATA15362647XXXXA5+,25143647XXXXA5-,151515473715AM,151526153715A7,1143332311
XXA#
134 DATA1144332311XXA#4,XX33234411XXA#5+,XX43332211XXA#5-,1122433311XXA#M
135 DATA1143113311XXA#7,1244342412XXB,1245342412XXB4,XX34244512XXB5+,XX44342312X
XB5-
136 DATA1223443412XXBM,1244123412XXB7,3535353513XXC6,4311221133XXCM6,3636363614X
XC#6
137 DATA4412231234XXC#M6,32002200XXXXD6,4513241335XXDM6,43113311XXXXD#6,46142514
36XXD#M6
138 DATA44123412XXXXE6,4715261537XXEM6,45133513XXXXF6,24133513XXXXFM6,46143614XX
XXF#6
139 DATA25143614XXXXF#M6,47153715XXXXG6,26153715XXXXGM6,111111133440#6,27163816
XXXXG#M6
140 DATA121212123445A6,4211322200XXAM6,131313133546A#6,4322333311XXA#M6,14141414
3647B6
141 DATA4423343412XXBM6,XX24133513XXCM7,3333331223XXC9,XX44331223XXC9+,XX2514361
4XXC#M7
142 DATA3434341324XXC#9,XX45341324XXC#9+,11112200XXXXDM7,3535351425XXD9,XX463514
25XXD9+
143 DATA32224311XXXXD#M7,3636361526XXD#9,XX47361526XXD#9+,33234412XXXXEM7,373737
1627XXE9
144 DATA43331122XXXXE9+,34244513XXXXFM7,33441223XXXXF9,44341223XXXXF9+,35254614X
XXXF#M7
145 DATA34451324XXXXF#9,45351324XXXXF#9+,36264715XXXXGM7,35461425XXXXG9,46361425
XXXXG9+
146 DATA37274816XXXXG#M7,36471526XXXXG#9,47371526XXXXG#9+,4311322200XXAM7,XX4514
351425A9
147 DATAXXX45351425A9+,1122113311XXA#M7,4131210011XXA#9,3142210011XXA#9+
148 DATA1223123412XXBM7,3232321122XXB9,XX43321122XXB9+
```

Simulazione di somma algebrica effettuata da microprocessore

QUANDO si lavora in BASIC spesso non ci si rende conto della complessità delle operazioni svolte dal computer. Dovrebbe esser noto che il microprocessore, anche per eseguire operazioni molto banali come PRINT 2 + 2, esegue, per obbedire al comando impartito, decine e decine di operazioni a velocità incredibile e lavora seguendo una logica matematica particolare che è quella "booleana".

Nel presente articolo si descriverà, appunto, il modo in cui il microprocessore esegue la somma tra due numeri in accordo alla logica di Boole.

Un computer "ragiona" solamente a base di 0 (= assenza di tensione) e di 1 (= presenza di tensione); qualsiasi istruzione o dato deve quindi essere "tradotto" utilizzando 1 oppure 0 prima di inserirlo all'interno della macchina. Poichè i numeri che possono essere trattati sono due (1,0) parleremo di notazione binaria.

La notazione a cui siamo abituati è però quella decimale costituita da dieci numeri (da 1 a 9 più lo 0). Vediamo allora in che modo è possibile trasformare un numero decimale, ad esempio 112, nel corrispondente binario.

Si divide il numero dato per due e si terrà conto se vi è resto o meno. Si continua quindi dividendo nuovamente il quoziente ottenuto per due, fino a che esso diventerà minore di due; in altre parole:

112 : 2 = 56 Resto = 0
56 : 2 = 28 Resto = 0
28 : 2 = 14 Resto = 0
14 : 2 = 7 Resto = 0
7 : 2 = 3 Resto = 1
3 : 2 = 1 Resto = 1

Il numero desiderato è pertanto 1110000 in cui il primo 1 a sinistra rappresenta il valore dell'ultimo quoziente e gli altri valori sono quelli dei resti ottenuti, letti dal basso in alto.

Il lettore verifichi le seguenti trasformazioni:

64 = 1000000
15 = 1111
256 = 100000000
255 = 11111111
254 = 11111110
1100 = 1110000100

Si presenta, però, un problema passando dalla teoria alla pratica: un sistema a microprocessore può memorizzare solo gruppi di otto simboli (0 oppure 1) alla volta perchè le memorie sono elementi fisici (circuiti integrati) costruiti per contenere, appunto, dei bytes (= gruppi di 8 simboli o bit) di lunghezza otto.

I numeri binari precedentemente calcolati, di lunghezza inferiore ad otto, possono allora essere "allungati" facilmente inserendo, a sinistra, tanti numeri 0 fino a raggiungere la lunghezza otto. Per-

tanto 1000000 diventerà 01000000 mentre 1111 si trasforma in 00001111 ecc. D'ora in poi, per facilità di lettura, scriveremo gli otto bit in due gruppi di quattro affiancati). E' come se noi, in pratica, scrivessimo 0138 invece di 138: sappiamo benissimo che lo zero posto prima di un numero non ha significato.

Come facciamo ora per i numeri più lunghi di otto bit?

Purtroppo non possiamo far nulla: non è possibile, almeno in questo primo stadio della trattazione, memorizzare un valore superiore a 255 (cioè: 1111 1111). Vi è inoltre un'altra limitazione: non è possibile trattare numeri negativi.

Sappiamo benissimo però che qualsiasi calcolatore tratta numeri positivi e negativi di ordine ben maggiore di 255.

Vediamo allora di eliminare le due limitazioni, cui prima accennavamo, iniziando dalla seconda.

E' noto che un numero negativo si differenzia da uno positivo per via del segno. Questo, che per noi è uno dei due simboli (più e meno) oltre i dieci dei numeri, deve però essere inserito in qualche modo nel computer come uno zero oppure un uno. Ebbene, se usiamo la convenzione per cui il primo bit a sinistra del byte interessato rappresenta il segno, sarà possibile trattare numeri positivi (0) e negativi (1) pagando ovviamente tale nuova possibilità con valori assoluti minori. Infatti è come se potessimo trattare non più gruppi di otto bit, ma di sette, dato che il primo ha il solo compito di rappresentare il segno, e con sette bit non si può rappresentare un valore maggiore di 127. Ricordiamo che, usando una convenzione seguita in tutto il mondo, il primo bit a destra è numerato con 0, mentre il primo a sinistra (cioè l'ottavo), è il bit N. 7, cioè: 7 6 5 4 3 2 1 0.

L'intervallo che possiamo trattare passa così da:

0 255 a -127 +128

Vediamo adesso perchè i due valori assoluti non sono eguali.

Se adottassimo la convenzione secondo cui la positività o negatività dipende solo dal segno (ottavo bit o bit N°7) avremmo ad esempio:

+2 = 0000 0010
-2 = 1000 0010
+64 = 0100 0000
-64 = 1100 0000

Ma se inoltre ha significato:
0000 0000 cioè zero,
il valore 1000 0000 significherebbe: -0
il che è assurdo perchè 0 è considerato solo positivo.

Per evitare tale errore si utilizza, per rappresentare numeri negativi, la cosiddetta notazione del complemento a due che esponiamo qui di seguito, senza dimostrazione.

Supponiamo di voler rendere negativo il numero quattro, che in binario è rappresentato dal numero 0000 0100.

Si cambia ciascun simbolo 0 con 1 e viceversa ottenendo:
1111 1011; si somma ora il valore 1 (o meglio: 0000 0001) a quello precedentemente trovato con risultato: 1111 1100. (Si ricordi che 1 + 1 = 0 con riporto di 1).

Il lettore verifichi che:
-64 = 1100 0000
-31 = 1110 0001
-1 = 1111 1111
-128 = 1000 0000

Prima di andare avanti è bene precisare in che modo eseguire le somme con la notazione binaria introducendo considerazioni sul CARRY (= riporto).

Ragioniamo per ora in decimale. Volendo eseguire la somma di 122+87 eseguiamo dapprima quella tra 7 e 2 e, dato che non vi è riporto, scriviamo 9; eseguendo successivamente 2+8, trascriviamo 0 e consideriamo il riporto 1 che sommeremo all'ultima cifra rimasta ottenendo in definitiva 209.

La stessa cosa faremo in binario considerando però il carry quando si sommerà 1+1. Ricorriamo ad un esempio:

```
______11____←  riporti
0001 0010 +
0000 0110 =
__________
0001 1000
```

```
180 REM ***
190 FORI=1TO39:G$=G$+"-":NEXT
200 PRINTG$:GOSUB330:A1=VL:A=VL:GOSUB260:PRINT"■+":A1$=A$
210 GOSUB330:A2=VL:A=VL:GOSUB260:PRINT"■=":PRINT"————————":A2$=A$
220 GOSUB370:GOSUB490:GOSUB590:GOTO200
230 END
240             REM *** ROUTINE DI CONVERSIONE
250             REM *** DA DECIMALE A BINARIO
260 A$="":FORI=1TO7:B=A/2:K$="0":IF(B-INT(B))THENK$="1"
270 IFI=5THENK$=K$+" "
280 A$=K$+A$:A=INT(B):NEXT
290 IFVL<0THENA$="1"+A$:GOTO310
300 A$="0"+A$
310 PRINT"■"A$;
320 RETURN
330 PRINTTAB(12);:INPUT"VALORE";VL:PRINT"■";
340 IFVL<-128ORVL>127ORVL>INT(VL)THEN330
350 RETURN
360 :           REM *** ROUTINE DI SOMMA ALGEBRICA
370 A3$="":V1=0:N=0:V=0:C1=0:FORI=9TO1STEP-1
380 A5$=MID$(A1$,I,1):IFA5$=" "THENA3$=" "+A3$:GOTO430
390 A=VAL(A5$):B=VAL(MID$(A2$,I,1)):IFI=2AND(A+B+C1)>1THENV1=1
400 IF(A+B+C1)>1THENGOSUB630:GOTO430
410 IFA=0ANDB=0ANDC1=0THENA3$="0"+A3$:GOTO430
420 A3$="1"+A3$:C1=0:REM DE SIMONE SOFTWARE/82
430 NEXT:PRINTA3$:PRINT:PRINT"CARRY="C1"   OVERFLOW=";
440 IF(V1=1ANDC1=0)OR(V1=0ANDC1=1)THENPRINT"1";:V=1:GOTO460
450 PRINT"0";
460 IFLEFT$(A3$,1)="1"THENN=1
470 PRINT"   NEGATIVO="N:PRINT:RETURN
480 :           REM *** ESAME CARRY OVERFLOW E NEGATIVO
490 IFC1=0ANDV=0ANDN=0THENPRINT"RISULTATO POSITIVO ESATTO=■";
500 IFC1=0ANDV=0ANDN=1THENPRINT"RISULTATO NEGATIVO ESATTO=■":
510 IFC1=0ANDV=1ANDN=0THENPRINT"RISULTATO IMPOSSIBILE"
520 IFC1=0ANDV=1ANDN=1THENPRINT"RISULTATO NEGATIVO DI OVERFLOW=■".
530 IFC1=1ANDV=0ANDN=0THENPRINT"RISULTATO POSITIVO ESATTO=■";
540 IFC1=1ANDV=0ANDN=1THENPRINT"RISULTATO NEGATIVO ESATTO=■";
550 IFC1=1ANDV=1ANDN=0THENPRINT"RISULTATO POSITIVO DI UNDERFLOW=■";
560 IFC1=1ANDV=1ANDN=1THENPRINT"RISULTATO IMPOSSIBILE"
570 RETURN
580 :           REM *** CALCOLO PER RISULTATI POSITIVI
590 J=0:AA=0:IFLEFT$(A3$,1)="1"THENN=1:GOSUB660:RETURN
600 FORI=9TO1STEP-1:A8$=MID$(A3$,I,1):IFA8$=" "THENNEXT
610 AA=AA+VAL(A8$)*2↑J:J=J+1:NEXT:PRINTAA:RETURN
620 :           REM *** RIPORTO
630 IFA+B+C1=3THENA3$="1"+A3$:RETURN
640 A3$="0"+A3$:C1=1:RETURN
650 :           REM *** CALCOLO PER RISULTATI NEGATIVI
660 J=0:AA=0:FORI=9TO1STEP-1:A8$=MID$(A3$,I,1):IFA8$=" "THENNEXT
670 IFA8$="1"THENA8$="0":GOTO690
680 IFA8$="0"THENA8$="1"
690 AA=AA+VAL(A8$)*2↑J:J=J+1:NEXT:PRINT"-"AA+1:RETURN
```

Partendo da destra abbiamo che 0+0=0; la seconda somma parziale è 1+1 che fornisce 0 (e non 2, dato che possiamo utilizzare solo i simboli 0 e 1) con riporto di uno che sommato a 0 e ad 1 (terza somma parziale), fornisce nuovamente 0 con carry di 1. Quest'ultimo è sommato a 0 e poi nuovamente a 0 (quarta somma parziale) fornendo zero come riporto. Le successive somme di coppie di valori relative ai bit 4, 5, 6, 7, non costituiscono problemi.

Ricordiamo che nel caso di somma contemporanea di più numeri binari si osserverà la norma seguente: se la "colonnina" di simboli (compresi eventuali riporti) è costituita solo da 0, si scrive 0 e si riporta 0; se il numero di 1 presenti è pari si scrive 0 e si riportano tanti 1 quanti ne rappresenta il numero di 1 presenti diviso per due; se il numero di 1 presenti è invece dispari, si scrive 1 e si riportano tanti 1, sulla "colonnina" successiva, quanti ne saranno rappresentati dalla parte intera della quantità di 1 divisa per due. Esempio:

```
_____1_____2° Riporto
_____1_111_1° Riporto
___________
0010 0111 +
0000 1111 +
0000 0100 =
___________
0011 1010
```

Vediamo ora di applicare la regola esposta nel seguente caso particolare:

```
1
0111 1111 | 127 +
0100 0000 |  64 =
__________|_____
1011 1111 | ? ? ?
```

Si è verificato un overflow: la somma dei due numeri induce un riporto nell'ottavo bit che è però quello rappresentante il segno e il risultato è errato benchè la regola della somma sia stata rigorosamente rispettata. Quest'ultima, infatti, è una regola matematica che non può tener conto delle "comodità" che noi introduciamo per tener conto dei numeri negativi. Pertanto si verificheranno errori quando la somma dei due numeri supera il valore +127 oppure è inferiore a -128.

Tali condizioni di errore sono simulate dal programma presentato in cui: C1 rappresenta il carry; N indica se il risultato ottenuto dalla somma è positivo o negativo (in pratica N si limita a leggere il valore del bit N°7: se esso è 1 vuol dire che è negativo e quindi N=1; in caso contrario N=0). L'ultima variabile, o meglio, *flag*, è V ed indica l'overflow. Tale flag sarà automaticamente posto ad 1 se, nell'eseguire la somma, vi è riporto tra il bit 6 ed il bit 7 ma non si verifica un contemporaneo riporto esterno (cioè riporto per una ipotetica colonnina N° 8). Sarà anche gettato ad 1 se non vi è riporto tra il bit 6 e il bit 7 ma vi è riporto esterno (vedi riga 440). Negli altri casi V sarà regettato e posto a zero.

Il flag V, tenuto conto dei flag N e C, è utilizzato proprio per tener conto di errori che si verificano a causa della convenzione usata per rappresentare il segno che può, come abbiamo visto; essere confuso per un valore numerico.

Facendo girare il programma il lettore può verificare quanto affermato studiando soprattutto le righe da 480 a 560. Le righe 510 e 560, in effetti, non saranno mai prese in considerazione dal programma e sono state trascritte per completare le combinazioni possibili.

Vediamo ora in che modo ampliare il campo dei numeri che è possibile sommare tra loro.

Sappiamo ormai che l'intervallo preso in esame è compreso tra i valori -128 e +127 e ciò a causa della lunghezza fisica del registro di memoria in cui vengono allocati i simboli 0 e 1. Se però noi potessimo disporre di memorie più "larghe" il problema sarebbe sicuramente di minore gravità. Possiamo, in questa ipotesi, considerare i numeri da trattare larghi non otto ma, ad esempio, sedici bit. Sarà pertanto possibile considerare (riservando, come prima, al primo bit a sinistra il significato di segno) l'intervallo di numeri interi compresi tra -32768 (cioè: 1111 1111 1111 1111) e +32767 (vale a dire: 0111 1111 1111 1111). Eseguendo la somma algebrica tra due numeri così rappresentati, il discorso sui flag V,N,C sarà valido non più esaminando il riporto tra il bit 6 ed il bit 7 e sul riporto

L'INCREDIBILE E'A PORTATA DI MANO.

PERSONAL COMPUTER

Commodore Italiana srl
Via F.lli Gracchi, 48
20092 Cinisello Balsamo (MI)
Tel. 02/6125651

Distributori Commodore:

LIGURIA - Pirisi Informatica
Piazza Cavour, 19 - 16043 Chiavari
Tel. 0185/30.10.31

PIEMONTE - Aba Elettronica di Caramia
Via Fossati, 5/C - 10141 Torino
Tel. 011/33.20.65

LOMBARDIA - Homic Personal
Computers srl - Piazza de Angeli, 3
20146 Milano - Tel. 02/49.88.201

VENETO, FRIULI-VENEZIA GIULIA,
TRENTINO-ALTO ADIGE
CO.R.EL. Friuli Computers
Via Mercatovecchio, 28 - 33100 Udine
Tel. 0432/29.14.66

EMILIA-ROMAGNA, MARCHE
S.H.R. srl - Via Faentina 175/A
48010 Fornace Zarattini (Ràvenna)
Tel. 0544/46.32.00

TOSCANA - M.C.S. Spa
Via Pier Capponi, 87 - 50132 Firenze
Tel. 055/57.13.80

UMBRIA - ALTO LAZIO
Atlas System srl
Via Guglielmo Marconi, 17 - 01100 Viterbo
Tel. 0761/22.46.88

LAZIO - Kiber Italia srl
P.le Asia, 21 - 00144 Roma Eur
Tel. 06/59.16.438

ABRUZZO - Pragma System srl
Via Tiburtina, 57 - 65100 Pescara
Tel. 085/50.883

CAMPANIA - Computer Market
Parco S. Paolo Isolato 9
80100 Napoli - Tel. 081/76.72.222

PUGLIA - Maselli x l'ufficio
Via L. Zuppetta, 5 - 71100 Foggia
Tel. 0881/76.1.11

PUGLIA - Business Automation Systems srl
Largo De Gemmis, 46/B-46/C-48-48/A-48/B
70124 Bari - Tel. 080/22.75.75-22.73.44

COMMODORE 64

INCREDIBILE IN TUTTO.

L. 825.000

Il prezzo.
Commodore 64 è il solo vero Personal Computer che costi meno di un milione, 825.000+IVA.
Il concorrente più vicino può darvi le stesse caratteristiche a non meno del doppio. Perchè Commodore produce solo Personal Computer e perchè li produce con il miglior rapporto prezzo-prestazioni oggi esistente.

Gli utilizzi.
Commodore 64 è ideale per mille utilizzi. Nel lavoro, in casa e nello studio. Le sue eccezionali qualità lo rendono adatto per l'uomo d'affari e per l'uomo di finanza, per il commerciante e per l'uomo di marketing, per la piccola industria e per i vari reparti di una grande azienda.
Serve per stime, bilanci, gestioni, magazzino, word processing. Moltissimi sono i programmi esistenti. In casa evade la corrispondenza, tiene conti e bilanci, comunica via telefono, suggerisce ricette, diete ed è ideale per chi vuole programmare. Nella scuola mette alla portata degli studenti uno strumento didattico di primissima importanza, con una vasta libreria di programmi e applicazioni.

Le caratteristiche.
64 K di memoria incorporata. 16 colori in tastiera. Alta risoluzione grafica. Effetto tridimensionale. Sintetizzatore musicale di livello professionale. Capacità di un secondo processore. Una gamma completa di periferiche e interfacce.

Provalo subito.
Da oggi Commodore 64 è presso i distributori autorizzati Commodore, i centri vendita GBC e presso i Bit Shop Primavera. Entra e provalo: scoprirai perchè è incredibile!

CALABRIA - Sirangelo Computers srl
Via Nicola Parisio, 25 - 87100 Cosenza
Tel. 0984/75.7.41

SICILIA - Edilcomput Progetti
dell'Ing. Giuseppe Carbone
Via La Farina, 141 Is. L - 98100 Messina
Tel. 090/29.28.269

SARDEGNA - S.I.I. - Sistemi
Integrati Informatica - Via S. Lucifero, 95
09100 Cagliari - Tel. 070/66.37.46

Centri vendita GBC

BIT SHOP PRIMAVERA vedere a pag.

ELEDRA 3 S
Milano, tel. 02/34.97.51
Roma, tel. 06/81.27.324 - Torino, tel. 011/30.99.101-2-3
Bologna, tel. 051/30.77.81 - Padova, tel. 049/65.54.88

software VIC 20

I nuovi prezzi bits & bytes:

VAS01N RICOCHET
Un gioco d'azione furiosa e di strategia in cui vi batterete contro il vostro Vic o contro un avversario umano. Livello automatico di difficolta', 5 varianti di gioco, azione, colore ed effetti sonori sbalorditivi. Espansione 8K. Lit. 40.000

VMI01N VIC VANGO
Una utility grafica che trasforma il vostro schermo in una "tela" elettronica. Il programma e' disegnato per solleticare l'immaginazione, la creativita' ed il coordinamento mano-occhio, il che ne fa' anche un ottimo strumento educativo. Joystick. Lit. 27.000

VMI02N ALIEN SOCCER
Una partita di calcio contro avversari alieni. Il gioco e' reso piu' vario e interessante dai "bumpers" sparsi nel campo, che vi danno punti quando colpiti, e da altre sorprese. Ore ed ore di divertimento! Joystick o paddle opzionali. Lit. 30.000

VMI03N GALACTIC CROSSFIRE
Un gioco d'azione spaziale in cui dovrete difendervi dall'imboscata dei micidiali robots Thorax e distruggerli. Dovrete anche guardarvi da tempeste di fuoco e da altre insidie mortali. Uno o due giocatori e 5 livelli di difficolta'. Lit. 30.000

VMI04N FRUIT FLY
Un gioco di destrezza manuale in cui dovrete aiutare un gruppo di agricoltori e di piloti di elicottero a distruggere un'orda di famelici parassiti che minacciano le coltivazioni. Joystick. Lit. 27.000

VMI05N CHECKBOOK
Un completo package di contabilita' bancaria consistente in 2 programmi: Checkwriter memorizza tutte le vostre operazioni ed effettua la quadratura per mese o per assegno; Checkreader e' un programma di ricerca, totalizzazione e visualizzazione di enorme utilita' nell'analisi del bilancio familiare. Opzione di stampa. Lit. 40.000

VTR01N GALACTIC BLITZ
Un gioco d'azione rapidissima: dovrete battervi con 15 diversi tipi di alieni, ciascuno con le sue caratteristiche. Disporrete di armi laser, fotoniche e faser. Alta risoluzione e linguaggio macchina ne fanno uno dei migliori giochi spaziali. Lit. 50.000

VUM01N VICALC
Un comodo programma di calcolo: i suoi 10 registri di memoria e 4 di dati sullo stack sono costantemente visualizzati sullo schermo. Premendo un singolo tasto, potrete ottenere i risultati di funzioni aritmetiche e matematiche, percentuali, numeri casuali, interessi composti e cosi' via. Questi risultati possono essere immagazzinati in qualunque registro di memoria. La precisione puo' essere scelta tra 0 e 9 cifre significative. Lit. 30.000

VUM02N ALIEN BLITZ
Un gioco d'azione in cui dovrete liberare il cielo da invasori alieni. Richiede riflessi prontissimi, specialmente ai livelli superiori di difficolta'. Comprende diversi "specials" per chi raggiunge alti punteggi o colpisce determinati obiettivi. Joystick. Lit. 50.000

VUM04N THE ALIEN 6K WITH JOYSTICK
Questa volta l'alieno siete voi, e dovrete difendervi, con trappole aeree gonfiabili ed altri espedienti, dagli abitanti di Critter City che cercano di annientarvi. Azione rapidissima ed effetti sonori. Espansione 3K. Lit. 50.000

VUM06N 3-D MAZE
Dovrete trovare la vostra strada attraverso le pareti ed i corridoi di un intricato labirinto. I vari passaggi si aprono realisticamente ai vostri occhi mentre cercate affannosamente una via d'uscita. Vi intratterra' per ore ed ore. Lit. 30.000

VUM09N VICAT
Il vostro data base personale: questo programma puo' gestire elenchi di nomi, indirizzi, numeri telefonici, ricette e molte altre cose. I records possono essere modificati, alfabetizzati, registrati o prelevati da nastro. Vicat e' particolarmente potente quando i suoi primi files vengono usati come directories. Ciascuno dei files (45 linee con 17 caratteri per linea) possono essere analizzati con una rapida funzione di ricerca. Espansione 3K o 8K. Lit. 50.000

VUM10N AMOK
I corridoi di Amok sono popolati da robots la cui sola finalita' consiste nell'annientare i visitatori indesiderati. Per salvarvi, dovrete prendere decisioni rapide e muovervi velocemente. I molti livelli di difficolta' di questo gioco vi daranno ore di divertimento. Richiede il joystick. Lit. 50.000

VUM10C AMOK
Il programma precedente su cartuccia. Lit. 80.000

VUM11N VITERM A
Un programma di utility che vi consentira' di connettervi telefonicamente con reti di comunicazione e di trasformare il vostro Vic in un terminale di computers piu' grandi. Richiede un modem con interfaccia RS-232. Lit. 40.000

VUM12N SUB CHASE
Una flotta di sommergibili sgancia mortali mine ad immersione. Per distruggerle, dovrete piazzare al posto giusto delle barriere di bombe di profondita'. Piu' riuscirete a distruggerne, tuttavia, piu' l'attacco nemico si fara' serrato! Espansione 8K. Lit. 50.000

VUM13N KOSMIC KAMIKAZE
La vostra astronave, in viaggio verso la Terra, verra' distrutta se anche uno solo degli invasori alieni la raggiungera'. La loro ammiraglia li protegge efficacemente e da essa dovrete guardarvi in special modo. Il vostro successo la spingera' a scatenare contro di voi nuove orde di invasori. Espansione 3K. Lit. 50.000

VUM14N VICHECK
Un programma per gestire la vostra contabilita' bancaria, compresi assegni, depositi e tutte le normali operazioni. Funzioni di controllo, di ricerca rapida, di aggiornamento e perfino di stampa assegni. Espansione 3K o 8K. Lit. 50.000

>>>>> N O V I T A' <<<<<

VTH01C VIC MUSIC COMPOSER	Lit. 87.000
VUM15N METEOR SHOWER	Lit. 30.000
VHE01N FUEL PIRATE	Lit. 34.000
VHE02N SIMON	Lit. 34.000
VHE03C TURTLE GRAPHICS	Lit. 80.000
VHE04N CONCENTRATION	Lit. 34.000
VHE05N VICTREK	Lit. 37.000
VAS02N RESCUE AT RIGEL	Lit. 60.000
VAS03N SWORD OF FARGOAL	Lit. 60.000
VTR02N SWARM!	Lit. 60.000
VTR03N SIDEWINDER	Lit. 60.000

Legenda - Ultima lettera del codice: N=Nastro C=Cartuccia.
Il presente listino annulla e sostituisce tutti i precedenti.
I prezzi si intendono al netto di IVA 18% e franco ns.sede di Milano.
Gli ordini possono essere inoltrati per lettera
o telefonando ai numeri (02) 573344 - 5468901.
La spedizione si effettua contrassegno, salvo accordi particolari.

Condizioni particolari ai Sigg. Rivenditori.

---

bits & bytes 20124 Milano Viale Monte Grappa 4 Tel. 02/573344-5468901-651628

esterno, ma sul carry tra il bit 14 ed il bit 15 e sul riporto dell'ipotetico sedicesimo bit.

Vediamo ora come passare dalla teoria alla pratica.

Per scrivere un numero in memoria lo porteremo (vedi schema) dapprima alla lunghezza di sedici bit inserendo, eventualmente, tanti 0 a sinistra quanti ne bastano. In seguito "spezzeremo" tale numero in due da otto bit ciascuno potendo così inserirli in due elementi da otto bit. Analogamente ci comporteremo con l'altro numero al quale vogliamo sommare il numero già memorizzato. Per eseguire, appunto, la somma, consideriamo le due metà di destra dei due numeri e le sommiamo *senza* tener conto del flag N e V, ma considerando *solo* il flag di carry C. E' ovvio che le due metà di destra così considerate in questa nuova convenzione, non sono limitate dai due valori +127 e -128 proprio perchè tale limitazione non ha senso dato che il bit N°7 non possiede il significato di segno.

Dopo aver eseguito la somma tra i corrispondenti otto bit di destra si sommano tra di loro i corrispondenti primi otto bit di sinistra, tenendo conto anche dell'eventuale carry della precedente somma parziale e prendendo in esame, stavolta, i tre flag V,C,N per controllare se vi è overflow (valori superiori a +32767) oppure underflow (valori inferiori a -32768).

Il programma presentato simula, in effetti, solo il caso di somme di numeri nel formato da otto bit.

Poichè il programma stesso è strutturato in una serie di subroutines di facile interpretazione, il lettore, per esercizio, può sofisticarlo aggiungendo alcune istruzioni per considerare l'intervallo -32768 +32767.

Si tenga comunque presente che per l'elaborazione di alcuni dati, come la notazione complemento a due, nel programma presentato si è fatto ricorso ad alcuni...trucchetti per evitare il calcolo in binario puro. Lo schema che completa il presente articolo rappresenta, infine, un utile riepilogo sull'argomento trattato.

A. De Simone

VIC REL

VOLETE controllare mediante VIC un sistema antifurto, un campanello o una sirena d'allarme, le porte del vostro garage, il lucchetto di sicurezza di casa vostra, l'accensione e lo spegnimento del sistema di riscaldamento, di lampade, trasmittenti, control-

lori remoti, valvole, pompe, telefoni, accumulatori, impianti di irrigazione, apparecchiature elettriche, temporizzatori, ventilatori, deumidificatori, ecc., ecc.? Quello che vi serve, allora, è un VIC-REL.

Questa cassetta contiene sei relay in uscita e due input tipo optoaccopiatori e si collega al VIC attraverso la porta utente (posta dietro il computer sulla sinistra).

Nell'esempio riportato nel completo manualetto d'uso, viene illustrata l'utilizzazione del VIC-REL per azionare automaticamente la porta d'ingresso (ed accendere una lampada di segnalazione) tutte le volte che qualcuno pressa il campanello di casa.

I conti di casa

Contabilità giornaliera, Dizionario, Ricettario, Agenda, Applicazioni matematiche e scientifiche: sono i titoli dei primi cinque pacchetti-prodotto per il Vic che verranno messi in commercio attraverso la rete di distributori Commodore per facilitare la diffusione di queste macchinette. Il prezzo di ciascun programma è di lire 50.000 (Iva inclusa) e comprende la cassetta/dischetto ed il manuale d'uso.

Il programma "Contabilità giornaliera", in particolare, vi dà la possibilità di registrare giornalmente le vostre entrate e le vostre uscite, dopo averle contrassegnate con un nome da voi scelto. Ad esempio: entrate: stipendi, eredità, vincete al totocalcio;

uscite: spese per il vitto, affitto, luce, gas, spese per l'automobile, ecc.

Tutte le descrizioni che potete dare delle vostre entrate e delle uscite vengono chiamate tecnicamente "conti".

In ogni gestione contabile, ed anche in quella che voi potete fare con questo programma, esistono dei conti di entrata e di uscita: questi avranno naturalmente anche un nome, come negli esempi fatti prima.

Lo strumento contabile non si ferma però qui, infatti all'interno di ogni conto di entrata o di uscita sono contenute altre descrizio-

ni, che costituiscono i cosiddetti sottoconti. Per fare un esempio: nel conto generale di entrata denominato "stipendi" potrete avere quali sottoconti: stipendio mamma, stipendio papà, stipendio figli, stipendio zio e così via. Alla stessa maniera nel conto generale di uscita denominato "spese generali" potreste avere i sottoconti: luce, gas, telefono, affitto, ecc.

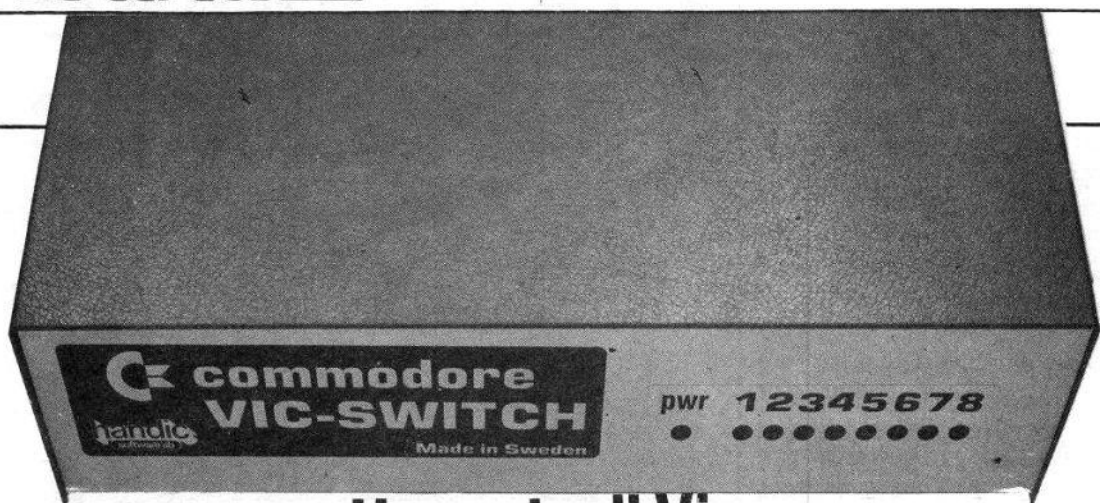
In questo modo potrete, giorno dopo giorno, registrare le spese (uscite) e le entrate in denaro nei rispettivi conti e sottoconti, ed avere in ogni momento sotto controllo la vostra situazione economica.

Il programma da, inoltre, la possibilità di chiamare i conti e i sottoconti nel modo che più vi fa comodo, di caricare mensilmente all'interno di ogni sottoconto le spese e le entrate, di farle visualizzare ogni volta che volete, di registrarle su nastro e conservarle indefinitamente.

Il programma, che al termine dell'uso, dà il saldo, cioè la differenza tra le entrate e le uscite, consente di gestire fino ad un massimo di 5 conti e 45 sottoconti.

ASSISTENZA VIC

Per facilitare la manutenzione del vostro VIC, la Commodore Italiana ha istituito presso la CATME di Milano (Via F. Severoli, 9 - tel. 02/4152840) un centro di assistenza a disposizione di tutti gli utenti.

Responsabile della Catme è il signor Fumagalli.

Una rete di Vic

COME si fa per far comunicare più utenti di VIC con la stessa apparecchiatura periferica (stampante, VIC 1515, memoria a dischetto 1540)?

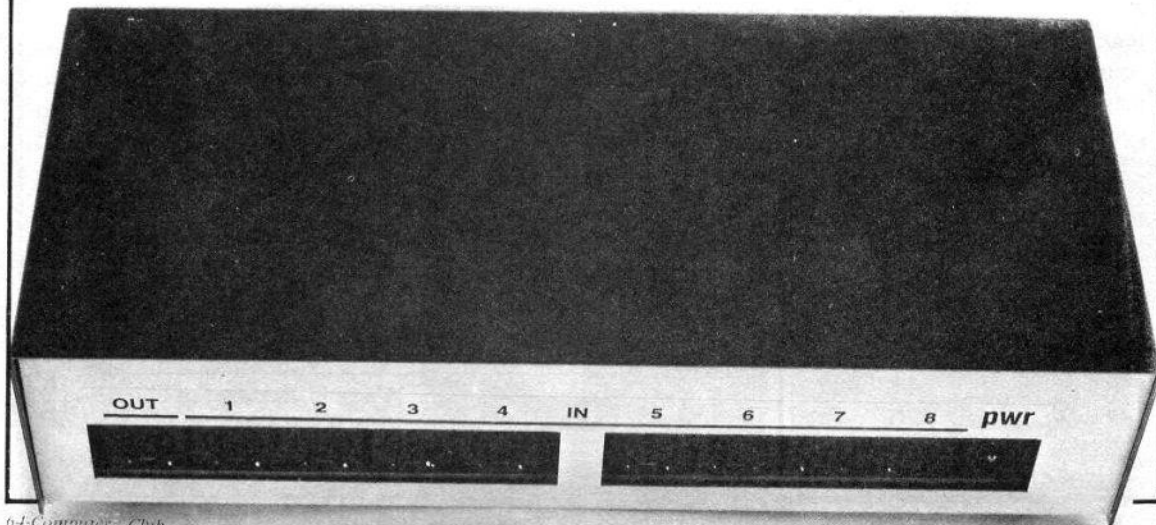
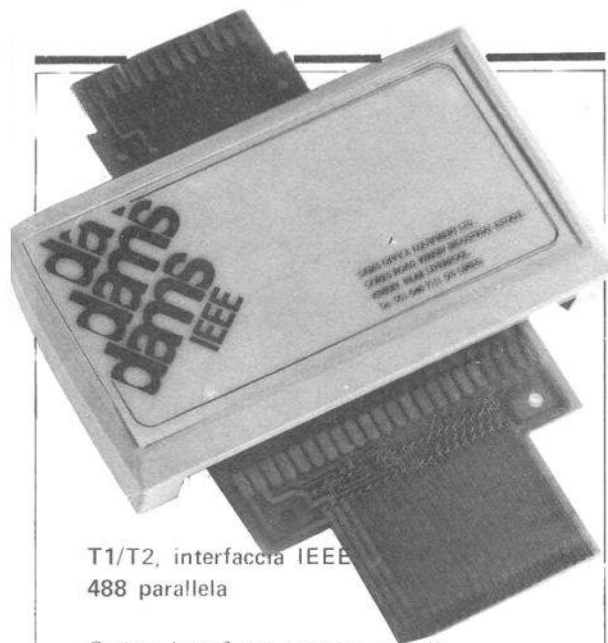
Occorre un VIC-Switch che si comporti come un vigile urbano e stabilisca le code di priorità tra i vari richiedenti l'accesso.

Con il VIC- Switch gli utenti possono accedere alle apparecchiature comuni in ordine: ad esempio, il n.ro 2 ha le priorità sull'utente n. 7, ma se in quel momento la linea è occupata dal VIC n.ro 6, non appena questo finisce, il nostro vigile, prima di dare il verde all'utente 2, cederà la linea al 7. In altri termini non esistono utenti privilegiati.

Come funziona

Nel pannello frontale notiamo 9 indicatori a LED. Quello più a sinistra, contrassegnato dalle lettere pwr (= power, corrente) indica se il VIC-Switch è acceso o spento. Gli altri 8 led indicano quale utente sta "transitando" per il switch in quel momento.

Il pannello posteriore comprende le prese DIN per le periferiche ed i computer da collegare.

Il cavo utilizzabile può essere lungo al massimo 12 metri: tanto quanto basta per una piccola rete locale.

T1/T2, interfaccia IEEE 488 parallela

Queste interfacce permettono di collegare il Vic 20 o il Commodore 64 a tutte le periferiche della serie CBM, come i floppy disk 4040, 8050 e 8250, la stampante 4022P, 8023P e CBM 6400.

Il vantaggio per l'utente è immediato: infatti, si ha finalmente la possibilità di avere a disposizione grosse quantità di memoria di massa, per la registrazione dei dati, e stampanti veloci e di qualità.

Il prezzo di queste interfacce è di lire 175 mila.

2004
BiP

VIC-20 un vero sistema con programmi sempre più nuovi

Matematico/scientifico. Alcune funzioni eseguibili: operazioni algebriche, logaritmi, fattoriali, seni e coseni, tangenti, equazioni di primo e secondo grado, radici.

Agenda rubrica telefonica. Per memorizzare sino a 90 nomi, indirizzi e numeri telefonici. Per programmare date e orari degli appuntamenti.

Archivio di parole. Con traduzione simultanea multilingue: inglese, tedesco, francese e un'altra lingua. Un vocabolario che si presta a mille altri utilizzi.

Ricettario. Un vero Chef personale che memorizza le ricette più segrete e le ricorda fornendo automaticamente le dosi necessarie.

Contabilità giornaliera. Utilissimo per registrare giornalmente entrate e uscite economiche: stipendi, rendite, vincite, spese per vitto, alloggio, auto, eccetera. Consente di tenere una facile contabilità mese per mese.

commodore
COMPUTER